AN INVESTIGATION OF A METHOD FOR OBTAINING A NUMERICAL INDICATOR OF PHASE COHERENCE

Robert Everett Beal

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

AN INVESTIGATION OF A METHOD FOR OBTAINING A NUMERICAL INDICATOR OF PHASE COHERENCE

by

Robert Everett Beal

June 1974

Thesis Advisor: G. A. Rahe

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN INVESTIGATION OF A METHOD FOR OBTAINING A NUMERICAL INDICATOR OF PHASE COHERENCE		5. TYPE OF REPORT & PERIOD COVERED Aeronautical Engineer's Thesis, June 1974 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert Everett Beal		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1974 13. NUMBER OF PAGES 136
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Fast Fourier Transform; Signal Processing; Phase Coherence; Amplitude Dependence; Normalization; Spectrum; Modified Fast Fourier Transform

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Fast Fourier Transform (FFT has dramatically changed the signal processing field by allowing the transformation of data from the time domain to the frequency domain at speeds rivaling and in many cases exceeding real time. The output spectrum obtained from the FFT depends on the amplitudes and the phase coherence of the sinusoids that comprise

Block 20 - ABSTRACT (Cont.)

the input waveform. It is the purpose of this investigation to modify the FFT so that it will rely more heavily on the phase coherence of these sinusoids rather than on their amplitude. In this way a second indicator of signal presence or absence would be available to add more reliability to the estimation process. The study proposes a method for reducing the dependence of the output coefficients on input amplitudes and investigates its merits with several types of data. A method of displaying the phase coherence indicators is presented.

An Investigation of a Method for Obtaining a Numerical Indicator of Phase Coherence

by

Robert Everett Beal
Lieutenant, United States Navy
B.S., United States Naval Academy, 1966
M.S., Naval Postgraduate School, 1973

Submitted in partial fulfillment of the
requirements for the degree of

AERONAUTICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
June 1974

ABSTRACT

The Fast Fourier Transform (FFT) has dramatically changed the signal processing field by allowing the transformation of data from the time domain to the frequency domain at speeds rivaling and in many cases exceeding real time. The output spectrum obtained from the FFT depends on the amplitudes and the phase coherence of the sinusoids that comprise the input waveform. It is the purpose of this investigation to modify the FFT so that it will rely more heavily on the phase coherence of these sinusoids rather than on their amplitude. In this way a second indicator of signal presence or absence would be available to add more reliability to the estimation process. The study proposes a method for reducing the dependence of the output coefficients on input amplitudes and investigates its merits with several types of data. A method of displaying the phase coherence indicators is presented.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure

I. INTRODUCTION

Signal processing deals with the problem of detecting signals in the presence of noise. If the noise is stationary Gaussian with a mean value of 0, and if the signal is absolutely stable, then it is theoretically possible to differentiate between the signal and the noise, no matter how small the signal to noise ratio. By integrating long enough (coherent integration) or averaging enough transforms (incoherent integration) the noise will eventually average out to zero or achieve a state of zero variance. This is the absolute ideal and is never found in practice. The noise is rarely, if ever, purely Gaussian, nor is it absolutely stationary. Signals are not absolutely stable, nor do they exist for infinite periods of time, thereby disallowing infinite integration or averaging times. Because of these real-world constraints there are limitations to the effectiveness of all signal processing methods when applied to practical problems.

The Fast Fourier Transform (FFT) is one of the most common and effective methods by which data can be transformed from the time domain to the frequency domain. This transformation produces a frequency spectrum that gives the relative power in the various frequencies that comprise the time domain data. Due to the practical limitations of signal processing stated above this method of signal processing is limited in effectiveness. It is the purpose of this study

to try to develop another transformation, similar to that of the FFT, that will produce a frequency spectrum that is more dependent on the phase coherence of the different frequency components that constitute the time domain data and less dependent on their power content. The spectrum analyst will then have a second indicator to aid him in making his determination of signal presence or absence. Hopefully this will result in more reliable estimates and allow signal detection in some situations that have previously been beyond the physical capabilities of the FFT.

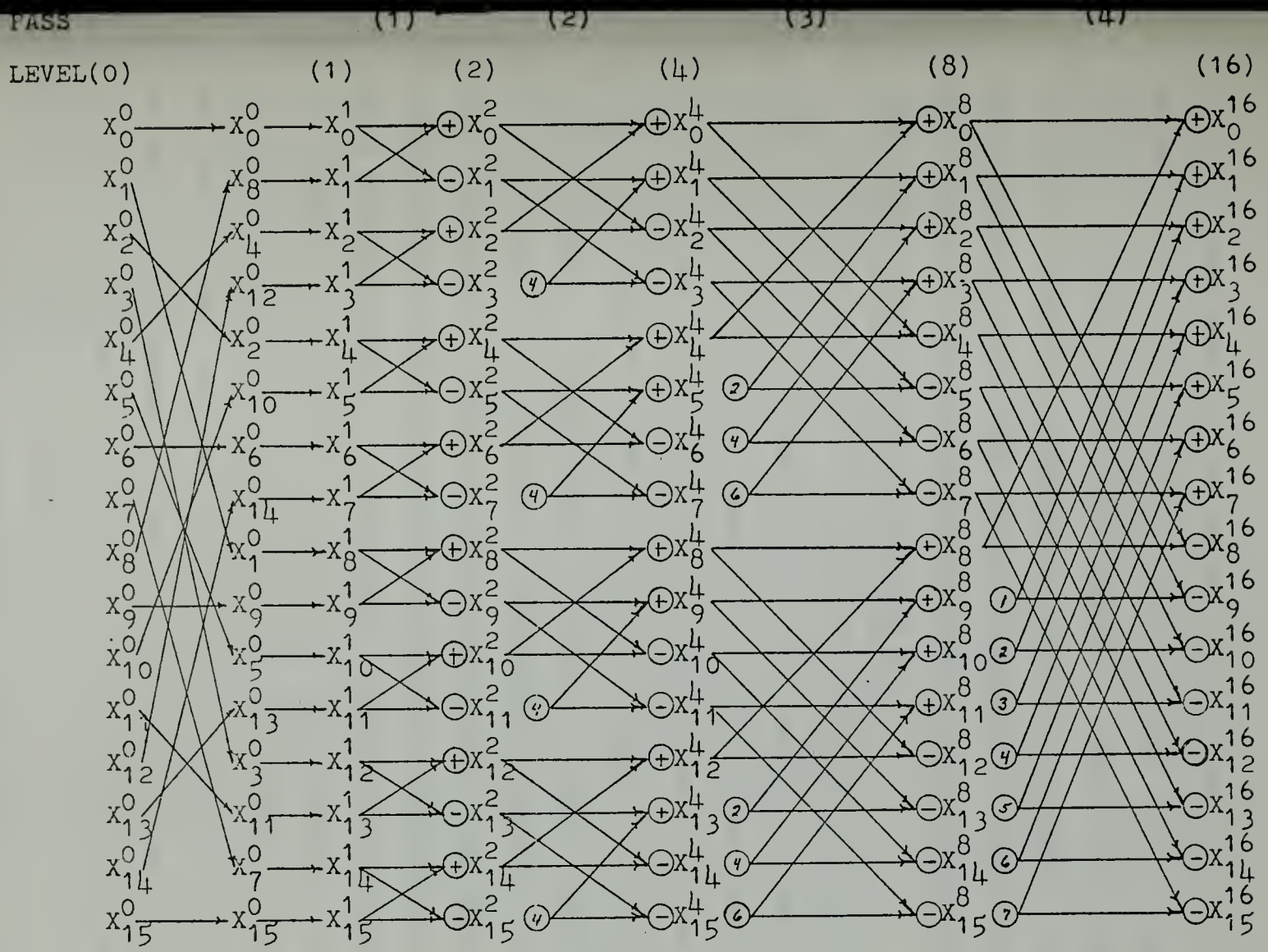
Part 1 of this investigation developes a procedure for obtaining a numerical indicator of phase coherence from the FFT. The procedure involves weighting the amplitudes of the components that add to produce each of the FFT output coefficients equally so the resulting spectrum will rely more heavily on the phase relationships between additive components than on their amplitudes. Part 2 compares the merits of the procedure's signal enhancement capabilities with that of the FFT. Part 3 summarizes these comparisons and Part 4 draws some conclusions about the differences between the Modified FFT's spectrum and that of the FFT.

II. PART 1 DEVELOPMENT

A. GENERAL

The Fast Fourier Transform (FFT) is an algorithm for computing the Fourier Coefficients of a discretely sampled waveform $X_T(t)$ where the subscript T implies X(t) to be finite in length (of length T) and also periodic in t with period T. $f_0 \equiv 1/T$ is the fundamental frequence of the FFT.

Appendix A gives a brief description of the Discrete and Fast Fourier Transforms. The flow graph of a 16 point FFT along with some of the notation to be used throughout this paper is presented in Figure 1-1. The term "level" will be used to denote the resultant vector formed after each pass (matrix multiplication) of the algorithm. Level (0) references the sampled version of the input waveform in its natural order. Level (1) references the input vector after bit reversal has taken place. Level (2) is the intermediate result (or vector) resulting from the first pass (matrix multiplication) through the FFT algorithm. The number 2 is used to imply that two elements of the input vector are involved in the computation of each element of the intermediate vector at level (2). The notation is similar for the remainder of the algorithm. A circle with a + or - in it implies addition or subtraction is to take place. A circle with a 1,2,3,... in it implies multiplication by W^1, W^2, W^3,... where $W \equiv \exp(-j2\pi/N)$.



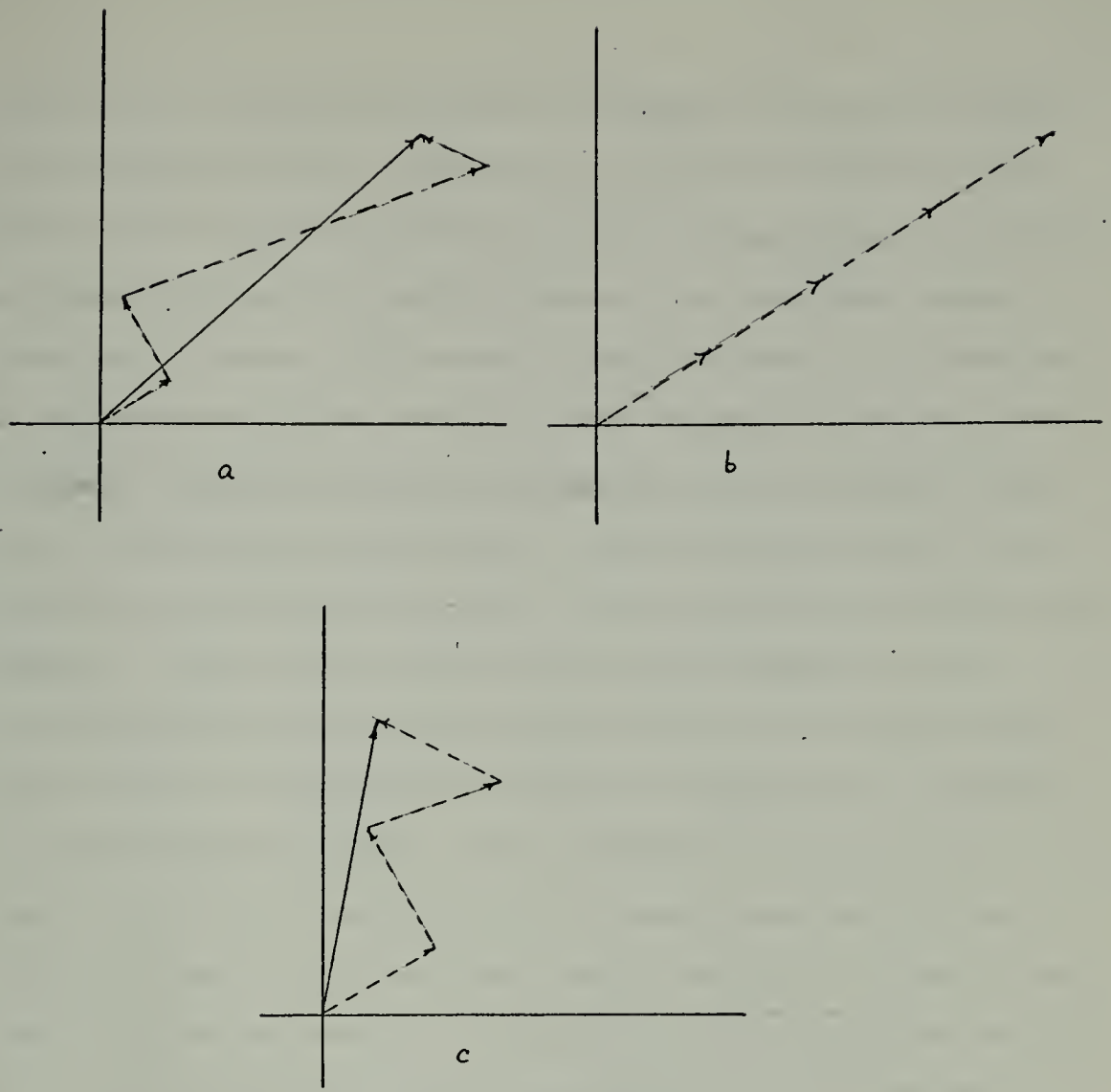
Flow Chart for 16 Point Fast Fourier Transform

Figure 1-1

The magnitudes of the output coefficients from the FFT depend on the amplitude of the input signal and its phase coherence throughout the transformation. When the FFT algorithm is viewed as a sequence of vector (or phasor) rotations and additions, it is the phase coherence of a signal that determines how the intermediate vectors will add in producing each output coefficient of the FFT.

Each pass through the algorithm consists of a sequence of vector rotations, followed by a vector addition. As previously stated, the length or magnitude of each output coefficient is a function of the magnitude of the intermediate vectors and of the manner in which they add to make up each output coefficient. The term phase coherence is used to describe the net effect of the phase relationships throughout the entire transform. A pure sinusoid would have phase coherence throughout the entire transform, whereas the phase relationship for noise would be completely random and incoherent. It is phase coherence that allows the intermediate vectors of a pure sinusoid to add end-to-end in a straight line, whereas the intermediate vectors for noise add at random phase angles, thereby producing output coefficients with an amplitude less than the maximum value they could achieve if they were to add in a straight line.

To give a better understanding of the concept of phase coherence consider the vector additions in Figure 1-2. The resultant vector may be viewed as the output vector of the FFT and its component vectors as the intermediate vectors of



Vector addition analogy of FFT

(a) Incoherent vector additions

(b) Coherent vector additions

(c) Incoherent vector additions, same as (a) but with intermediate components normalized to same length.

Figure 1-2

the FFT that have been rotated and added to form the resultant or output vector. Figure 1-2a could be viewed as incoherent noise with no particular order in the manner in which the component vectors add to produce the resultant vector. Conversely, Figure 1-2b shows a resultant vector of what may be interpreted as the result of the transformation of a pure sinusoid with zero phase or frequency shift throughout the time record being transformed. Each component adds to the others in a straight line and at the phase angle of the input sinusoid. The vector in 2a achieves its magnitude not through phase coherence but simply because one of its component vectors dominates the others in magnitude. If each of its component vectors were restricted to having unit length while still maintaining its same phase angle, then the magnitude of the resultant vector would be forced to depend on phase coherence (the ability of the component vectors to add in a straight line) throughout the transform and not on the magnitude of any of its component vectors. This situation is depicted in Figure 1-2c.

When the output of the discrete correlation performed by the FFT is squared, the resulting spectrum is proportional to the power contained in the original waveform at each of the correlating frequencies n/T (n=0,1,2,...,N/2-1) for real valued input waveforms and is referred to as the power spectral density of the input waveform. The association of the output coefficients of the FFT is strongly related to, and dependent on, the power of the input waveform at its various

frequencies, or more explicitly, at the correlation frequencies of the FFT algorithm.

It would seem plausible that if the magnitude of the intermediate vectors could be removed then the output spectrum would rely more on the sinusoidal nature of the input waveform than on the power of the input sinusoid. Or conversely, the ability of large vectors, due to incoherent noise, to corrupt or mask a coherent sinusoid would be reduced. The objective is to develop an indicator of the presence of a signal through phase coherence with as little emphasis as possible on signal or noise power, an indicator which could be used in conjunction with the FFT to aid in estimating signal presence or absence.

B. PREMISE

The basic premise of this study is to modify the FFT algorithm in such a manner that the output coefficients give an indication of the relative phase coherence of the sinusoids that make up the input waveform. To test the modified transform to see if it meets the premise, sinusoids that complete an integral number of cycles across the time record will be considered, that is, sinusoids that are integral multiples of the fundamental frequency of the transform f_0. These are of course the correlating sinusoids of the FFT algorithm. It will be required to show that the phase angle of the output coefficient corresponding to a pure sinusoidal input at any frequency $nf_0 (n=1,2,\ldots,N/2-1)$ reflects the

phase angle of the input sinusoid. This must be true if the intermediate vectors are to add in a straight line for a phase coherent sinusoid.

If the resulting output coefficient representing an input sinusoid of frequency nf_0 reflects the phase angle of an input sinusoid at these frequencies, then the premise has been satisfied. Sinusoids at frequencies other than integral multiples of f_0 are seen by the correlation process as a summation of sinusoids at the frequencies nf_0. That is, the FFT output does not reflect the phase angle of the input sinusoids at frequencies other than integral multiples of f_0 but instead approximates these sinusoids by sinusoids that are at integral multiples of f_0. Therefore the phase angles obtained do not reflect the phase angle of the input sinusoid, but instead reflect the phase angles of the correlating sinusoids that, when summed, will approximate the input waveform. Therefore it is meaningful to test the premise on input sinusoids that are integral multiples of f_0.

C. IMPLEMENTATION OF THE MODIFICATION

The output coefficients of the FFT are in general complex numbers of the form $c=a+jb$ with magnitude $|c| = (a^2+b^2)^{\frac{1}{2}}$ and phase angle $\theta=\text{Tan}^{-1}(b/a)$. When an output coefficient of the FFT is viewed with the objective of determining the phase angle of its corresponding input sinusoid, the phase angle obtained from $\theta=\text{Tan}^{-1}(b/a)$ is the average difference in phase angle between the input sinusoid and a reference cosine. If

a sine wave is used for the reference sinusoid then the phase angle $\theta=\text{Tan}^{-1}(a/b)$ must be used. In addition, one must know whether the correlating sinusoids were of the form $\cos(2\pi nk/N) - j\sin(2\pi nk/N)$ or $\cos(2\pi nk/N)+j\sin(2\pi nk/N)$. Either form results in the same magnitude but with a different phase angle. When θ is computed as $\text{Tan}^{-1}(b/a)$, changing the sign of the sine wave correlation results in a reflection of the phase angle across the real axis. When θ is computed as $\text{Tan}^{-1}(a/b)$, the sign change results in a reflection across the imaginary axis. The phase angle has physical meaning if one recognizes what the reference sinusoid was and whether $\exp(-j2\pi nk/N)$ or $\exp(j2\pi nk/N)$ was used for the correlation process.

After each pass of the FFT algorithm (at each level), the intermediate coefficients can likewise be represented in the same manner as the output coefficients. These intermediate coefficients are analogous to the intermediate vectors referred to in Figure 1-2. The object will be to make these intermediate coefficients unit vectors, vectors of magnitude 1. In this way the phase relationships of these vectors will dominate the length of any given output coefficient with as little emphasis on the magnitude of these vectors as possible. The method utilized will be simply to divide each coefficient by its magnitude. This process will be referred to as normalization since it normalizes the magnitude of the coefficients to a value of 1 and in no way alters the ratio of the real to imaginary part of each coefficient, that is, the phase angle of each coefficient is retained.

The problem then becomes one of determining where the normalization should take place in the FFT algorithm. The following properties of the FFT butterfly will be used to make this determination for real valued input waveforms:

(1) All of the coefficients at the 2 level are real valued since each coefficient at this level is either the sum or difference of 2 of the original real valued input coefficients.

(2) Half of the coefficients at the 4 level are complex numbers, namely the odd numbered coefficients X_k^4 (k=1,3,5,..., N-1) and these can have any phase angle, the rest are real valued.

(3) The odd numbered coefficients at the 4 level affect only the odd numbered output coefficients (those associated with f_0 and its odd harmonics), and similarly, the even numbered coefficients affect only the even numbered output coefficients (those associated with the even harmonics of f_0).

(4) Above the 4 level an increasing number of coefficients become complex, for example at the 8 level the complex coefficients are: X_k^8 (k=1,2,3,5,6,7,9,10,11,...,N-1). At the 16 level the complex coefficients are: X_k^{16} (k=1,2,3,4,5,6,7, 9,10,11,12,13,14,15,...,N-1), etc.

If the input coefficients are normalized, the only possible phase angles that they can have are 0 or 180 degrees (each with a magnitude of 1) since they are all real numbers. With N input coefficients then there will only be 2^N possible normalized input vectors (if the value zero is allowed then

there will be 3^N possible input vectors). This results in a finite number of output phase angles and thus by contradiction prohibits the vectors from adding in such a manner as to reflect all possible input phase angles.

The same argument is true for the 2 level where the coefficients again are real numbers. In fact the number of possible output phase angles is less than when the coefficients were normalized at the input. The number of different vectors is still 2^N at the 2 level but there is one less pass between the 2 level and the output than between the input and the output, thereby reducing the number of passes in which coefficients are combined.

The problem of determining where to implement this normalization procedure is greatly simplified if one remembers that the purpose of this modification is to achieve vector additions of the intermediate coefficients so that these vectors will add in such a way that the resultant vector is oriented in the direction of the input waveform's phase. In general that phase will be measured with respect to a cosine waveform throughout this study.

What the above suggests is that a coefficient should be complex before it is normalized in order not to restrict the total number of phase angles achievable by the output coefficients.

The 4 level is the first level where complex numbers arise. As previously stated half of the coefficients affect only the output coefficients related to even harmonics of f_0.

At this level the transform can be split up into two distinct and independent parts, one for the odd harmonics and one for the even harmonics. The real coefficients are the building blocks of the even harmonics, by the previous argument of a finite number of output phase angles it will not be possible to normalize these coefficients and obtain all possible phase angles at the output.

The complex coefficients at this level are the intermediate coefficients that make up the output coefficients relating to the odd harmonics of f_0. If these coefficients are normalized the output coefficients corresponding to the odd harmonics of f_0 will reflect the phase angle of any input sinusoid at the frequencies $f_0, 3f_0, 5f_0, \ldots, (N/2-1)f_0$.

Demonstration:

Consider the bit reversed form of a 16 point transform of some function x(t) and the FFT butterfly for these coefficients up to the 4 level in Figure 1-3. As the notation indicates each coefficient at the 4 level is made up of four elements of the input vector. Each grouping of four coefficients at the 4 level can be rewritten as:

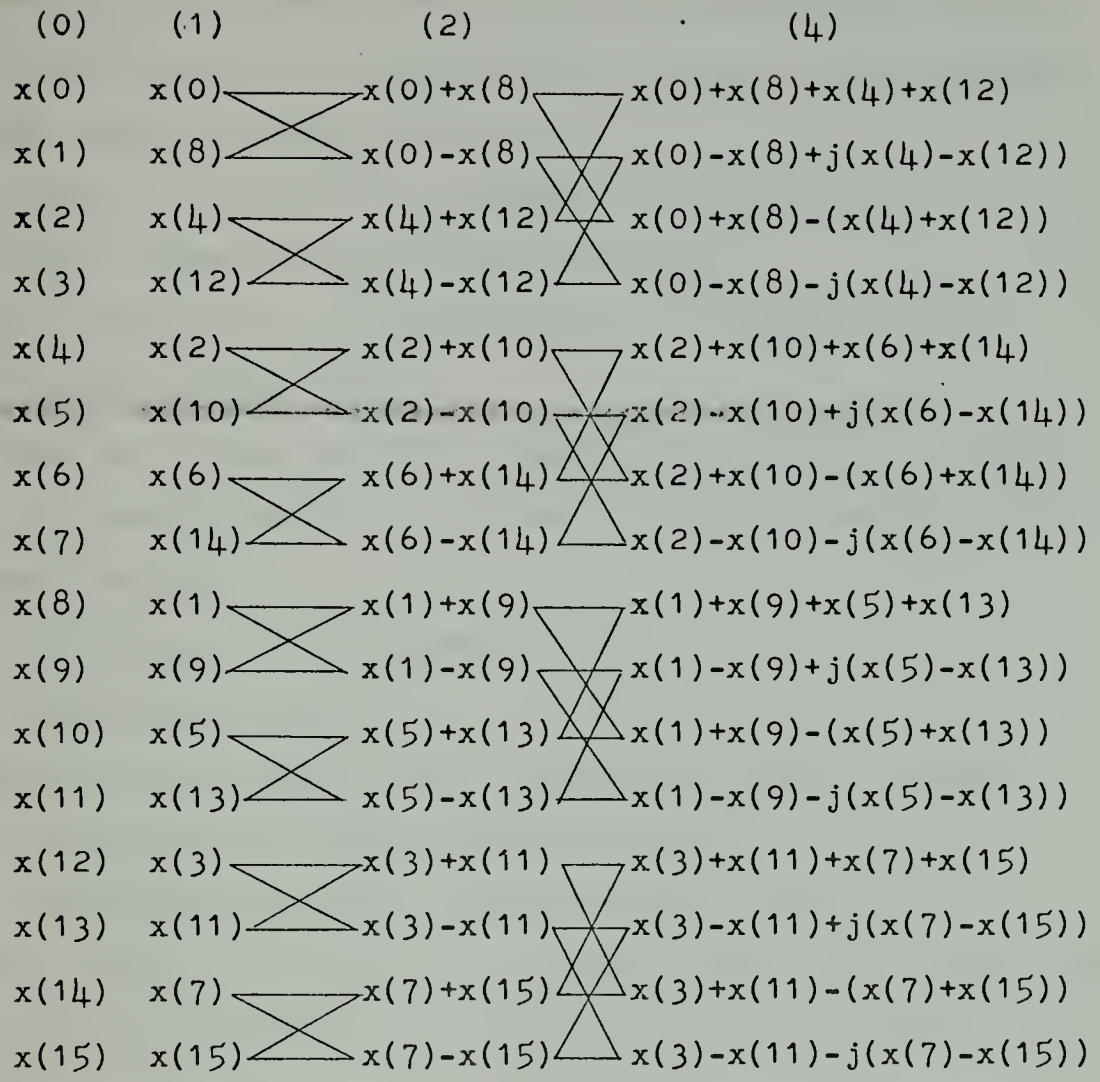
$$X^4(4j) = x(0+j) + x(N/2+j) + x(N/4+j) + x(3N/4+j)$$

$$X^4(4j+1) = x(0+j) - x(N/2+j) + j[x(N/4+j) - x(3N/4+j)]$$

$$X^4(4j+2) = x(0+j) + x(N/2+j) - [x(N/4+j) + x(3N/4+j)]$$

$$X^4(4j+3) = x(0+j) - x(N/2+j) - j[x(N/4+j) - x(3N/4+j)]$$

where $j=0,1,2,\ldots,N/4-1$. This is a valid generalization for any N. The coefficients in question are the complex coefficients and only these coefficients will be considered. Since



The FFT algoritm for a 16 point transform up to the 4 level

Figure 1-3

normalization is simply the division of any coefficient by its magnitude, then the normalization factor (the magnitude of the complex coefficients) is:

$$\{[x(j) - x(N/2+j)]^2 + [x(N/4+j) - x(3N/4+j)]^2\}^{\frac{1}{2}}$$

The input sinusoid can be generalized as:

$x(t) = A\sin(2\pi n f_0 t + \theta)$ where $n=1,3,5,\ldots,N/2-1$, $f_0=1/T$ and θ is arbitrary $-\pi \leq \theta \leq \pi$.

The sampled version of $x(t)$, namely $x(k)$ $(k=0,1,2,\ldots,N-1)$ is: $x(k)=A\sin(2\pi n f_0 k\Delta T + \theta)$ where ΔT is the time increment between samples.

Now: $T=N\Delta T$ and $f_0=1/T=1/(N\Delta T)$, then:

$$x(k)=A\sin(2\pi nk/N + \theta).$$

Therefore the divisor of the coefficients can be written,

$$\{[A\sin(2\pi nj/N+\theta)-A\sin(2\pi n(N/2+j)/N+\theta)]^2$$
$$+[A\sin(2\pi n(N/4+j)/N+\theta)-A\sin(2\pi n(3N/4+j)/N+\theta)]^2\}^{\frac{1}{2}}$$

letting $C=2\pi nj/N$ and $B=2\pi n$, then the divisor can be rewritten:

$$A\{[\sin(C+\theta)-\sin(C+B/2+\theta)]^2 + [\sin(C+B/4+\theta)-\sin(C+3B/4+\theta)]^2\}^{\frac{1}{2}}$$

Now, $\sin(x+y)=\sin(x)\cos(y)+\sin(y)\cos(x)$

Expanding the above,

$$A\{[\sin(C)\cos(\theta)+\sin(\theta)\cos(C)-\sin(C)\cos(B/2+\theta)-\sin(B/2+\theta)\cos(C)]^2$$
$$+[\sin(C)\cos(B/4+\theta)+\sin(B/4+\theta)\cos(C)$$
$$-\sin(C)\cos(3B/4+\theta)-\sin(3B/4+\theta)\cos(C)]^2\}^{\frac{1}{2}}$$

Now, $\cos(x+y)=\cos(x)\cos(y)-\sin(x)\sin(y)$

Expanding once more,

$$A\{[\sin(C)\cos(\theta)+\sin(\theta)\cos(C)-\sin(C)\cos(B/2)\cos(\theta)$$
$$+\sin(C)\sin(B/2)\sin(\theta)-\cos(C)\sin(B/2)\cos(\theta)$$
$$-\cos(C)\sin(\theta)\cos(B/2)]^2 + [\sin(C)\cos(B/4)\cos(\theta)$$
$$-\sin(C)\sin(B/4)\sin(\theta)+\cos(C)\sin(B/4)\cos(\theta)$$
$$+\cos(C)\sin(\theta)\cos(B/4)-\sin(C)\cos(3B/4)\cos(\theta)$$
$$+\sin(C)\sin(3B/4)\sin(\theta)-\cos(C)\sin(3B/4)\cos(\theta)$$
$$-\cos(C)\sin(\theta)\cos(3B/4)]^2\}^{\frac{1}{2}}$$

Now $B=2\pi n$ $(n=1,3,5,\ldots,N/2-1)$

$=2\pi,6\pi,10\pi,14\pi,\ldots$

then $B/2=\pi,3\pi,5\pi,7\pi,\ldots$

and $\sin(B/2)=0$

$\cos(B/2)=-1$

$B/4=\pi/2,3\pi/2,5\pi/2,7\pi/2,\ldots$

$\sin(B/4)=\pm 1=(-1)^{[(n+1)/2]+1}$

$\cos(B/4)=0$

$3B/4=3\pi/2,9\pi/2,15\pi/2,21\pi/2,\ldots$

$\sin(3B/4)=\mp 1=(-1)^{(n+1)/2}$

$\cos(3B/4)=0$

Making the above substitutions:

$$A\{[\sin(C)\cos(\theta)+\sin(\theta)\cos(C)+\sin(C)\cos(\theta)+\cos(C)\sin(\theta)]^2$$
$$+[\mp\sin(C)\sin(\theta)\pm\cos(C)\cos(\theta)\mp\sin(C)\sin(\theta)\pm\cos(C)\cos(\theta)]^2\}^{\frac{1}{2}}$$
$$=A\{[2\sin(C)\cos(\theta)+2\sin(\theta)\cos(C)]^2+[\mp 2\sin(C)\sin(\theta)$$
$$\pm 2\cos(C)\cos(\theta)]^2\}^{\frac{1}{2}} = A\{4\sin^2(C)\cos^2(\theta)+8\sin(C)\cos(\theta)\sin(\theta)\cos(C)$$
$$+4\sin^2(\theta)\cos^2(C)+4\sin^2(C)\sin^2(\theta)-8\sin(C)\sin(\theta)\cos(C)\cos(\theta)$$
$$+4\cos^2(C)\cos^2(\theta)\}^{\frac{1}{2}} = 2A\{\sin^2(C)[\cos^2(\theta)+\sin^2(\theta)]$$
$$+\cos(C)[\sin^2(\theta)+\cos^2(\theta)]\}^{\frac{1}{2}} = 2A\{\sin^2(C)+\cos^2(C)\}^{\frac{1}{2}} = 2A$$

Or, the divisor at the 4 level for the complex coefficients is dependent only on the amplitude of the input sinusoid; it is independent of N,n,j, or θ. This is equivalent to dividing all of the coefficients by a constant for any given input sinusoid of the form $x(t)=A\sin(2\pi nf_0t+\theta)$ and in no way alters the phase angle of the output coefficients. Its net effect is that of scaling the input sinusoid such that it is replaced by:

$x(t)=0.5\sin(2\pi nf_0t+\theta)$ for any amplitude A.

It can likewise be shown that a similar normalization at the 8 level results in division of the coefficients by 4A, at the 16 level the divisor is 8A and in general the divisor will be (L/2)A where L is the level at which the normalization was performed.

The second half of the transformation can not be normalized at the 4 level for reasons already stated. At the 8 level the coefficients once more divide, that is, the transformation can be broken down into 4 independent transformations, each part being related to one-fourth of the output coefficients. The intermediate coefficients 1,5,9,13,.... make up their corresponding output coefficients, likewise 3,7,11,15,... make up their corresponding output coefficients. These coefficients offer no problem since they were already complex at the 4 level. Coefficients 0,4,8,12,... are still real valued at the 8 level and do not lend themselves to normalization at this point. Coefficients 2,6,10,14,... were not complex at the 4 level but are complex at the 8

level. A similar proof to the one used for normalization of the intermediate coefficients of the odd harmonics of f_0 at the 4 level can be used to show that normalizing the coefficients 2,6,10,14,... at the 8 level does not alter the phase angle of the output coefficients corresponding to these frequencies for sinusoidal inputs that are integral multiples of f_0. Similarly at the 16 level the intermediate coefficients that make up the frequencies nf_0 ($n=4,12,20,28,\ldots$) become complex and can therefore be normalized without altering the phase angle of the output coefficients corresponding to these frequencies.

This process continues until one reaches the output of the FFT at which time all of the coefficients (except F(0), the D.C. term) have become complex and therefore normalization could have been performed. The idea of normalizing different coefficients at different levels produces one serious drawback. Normalization fixes the maximum magnitude an output coefficient can achieve. This is due to the fact that every intermediate coefficient somewhere in the FFT algorithm has been replaced with a magnitude of 1. Knowing where the normalization occurred one then knows how many more passes will be performed before the algorithm is complete. Since each pass consists of rotating and then adding coefficients, the number of vector additions remaining per output coefficient is then equal to the number of remaining passes. Since the coefficients have magnitude 1, they can achieve a magnitude at the output no larger than 2^P where P is the number of

remaining passes to be performed. For instance, in a 32 point FFT, if the intermediate coefficients that make up the odd harmonics of f_0 are normalized at the 4 level there will be three remaining passes yet to be performed. At the 8 level their maximum magnitude is 2, at the 16 level it is 4 and at the 32 level (the output in this case) their magnitude can be 8. The only way the value 8 can be achieved is if the input was in fact a pure sinusoid at a frequency nf_0 (n=1,3, 5,...). It is important to note that if one knows how many points (N) were in a transformation, and what frequency a particular output coefficient is related to, that is, if one knows N, then the maximum value that coefficient can achieve is also known. From this information one may be able to make a prediction as to the probability of the existence of a signal at the frequency. If one also considers the magnitude of this coefficient from more than one transform, then his prediction becomes more reliable. This in itself could be a powerful tool in signal processing work since the output coefficients are independent of the amplitude of the input signal.

The serious drawback occurs when one analyzes a spectrum where the intermediate coefficients have been normalized at different levels. It is a simple procedure to multiply each output coefficient by the number of the level at which it was normalized. In the previous example a 32 point transform was considered and it was shown that the maximum magnitude that an output coefficient, dependent on normalization of

its intermediate coefficients at the 4 level, could achieve was 8. Then 8 times 4 = 32. Likewise an output coefficient dependent on normalization of its intermediate coefficients at the 8 level can achieve a magnitude of 4. Then 4 times 8 = 32. In general, the magnitude of the output coefficients can be made a constant (namely N) for an input sinusoid of the form $A\sin(2\pi n f_0 t+\theta)$ $(n=1,2,3,\ldots,N/2-1)$ independent of A, n, and θ. And the phase angle θ can be obtained from the output coefficients.

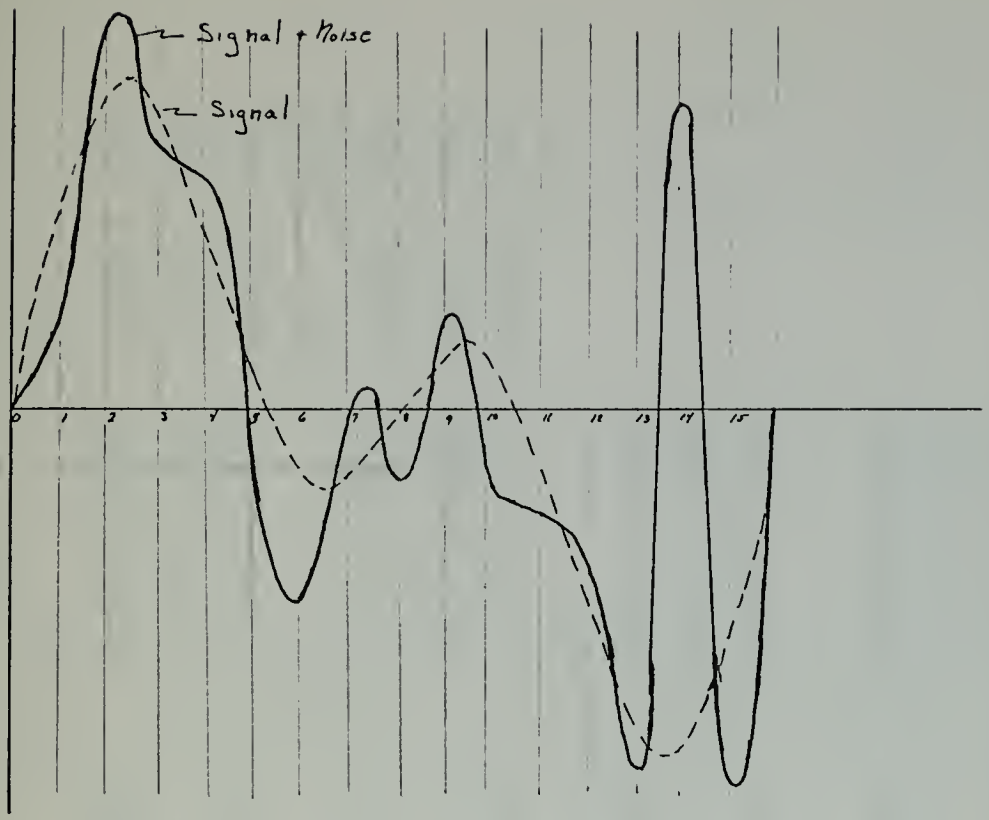
When the input waveform consists of signal plus noise, the multiplication process leads to erroneous results. To best explain this problem, consider the output coefficient which is dependent on normalization at the N^{th} level. This will always be the frequency $(N/4)f_0$ since this frequence coefficient does not become complex until after the last pass of the FFT algorithm has been performed. This coefficient will always have a magnitude of 1 after normalization and it will have a magnitude N after it has been multiplied by its normalization level, which is always N in this case. That is, the frequency $(N/4)f_0$ will always have the magnitude N regardless of whether there was signal or noise at that frequency. To a lesser degree this same problem extends back to the frequencies that were normalized at the N/2 level, specifically $(N/8)f_0$ and $(3N/8)f_0$. In general the frequencies that are normalized late in the transform tend to dominate the spectrum. This can be explained if one considers that the multiplication factor was simply an artificial means

of adding amplitude to the output coefficients so that all of the frequencies could achieve the same maximum value. Frequencies that have their components normalized early in the transform rely more heavily on phase coherence to achieve large amplitudes. The frequencies that have their components normalized late in the transform have less time to achieve large amplitude through phase coherence.

One solution to the above problem is to compare coefficients that are dependent upon normalization at a given level independent of the remaining coefficients. This has the drawback that the reliability of the estimate of signal presence is reduced since the comparison is made against fewer coefficients. In addition it is inconvenient and lacking in continuity since at best every other coefficient is compared leaving gaps between frequencies.

A second solution was devised in which all of the coefficients that depend on normalization at a given level are summed. Their mean value is determined and then each coefficient is divided by this mean value. This has the effect of giving the coefficients dependent on each level of normalization a mean value of 1. The spectrum can then be recombined with the effect of giving the entire spectrum a mean value of 1. A quick example will illustrate this procedure.

Consider a 16 point FFT with an input signal consisting of equal amplitude 1 hertz and 2 hertz sinusoids plus noise. Figure 1-4 shows the input sinusoids and the sinusoids plus noise. Figure 1-5 shows the FFT algorithm and Figure 1-6



	Signal	Signal + Noise
0	0.0	0.1
1	1.0898	0.6
2	1.7071	2.0
3	1.6310	1.4
4	1.0	1.2
5	0.2168	-0.3
6	-0.2929	-1.0
7	-0.3244	0.0
8	0.0	-0.4
9	0.3244	0.5
10	0.2929	-0.4
11	-0.2168	-0.5
12	-1.0	-0.9
13	-1.6310	-1.9
14	-1.7071	1.6
15	-1.0898	-2.0

1 hz + 2 hz sinusoids + noise

Figure 1-4

FFT

Level / Coeff	(0)	(1)	(2)	(4)	(8)	(16)	|16|	
0	0.1	0.1	-0.3	0.0	2.2	0.0	0.0	
1	0.6	-0.4	0.5	0.5+j2.1	4.0355+j1.9585	2.3949+j5.9958	6.4564	1hz
2	2.0	1.2	0.3	-0.6	-0.6 +j1.0	-0.3172+j5.3840	5.3933	2hz
3	1.4	-0.9	2.1	0.5-j2.1	-3.0355-j2.2415	-4.0398-j1.6407	4.3603	3hz
4	1.2	2.0	1.6	2.2	-2.2	-2.2	2.2	4hz
5	-0.3	-0.4	2.4	2.4-j2.6	-3.0355+j2.2415	-2.0312+j2.8423	3.4935	5hz
6	-1.0	-1.0	0.6	1.0	-0.6 -j1.0	-0.8828+j3.3840	3.4973	6hz
7	0.0	1.6	-2.6	2.4+j2.6	4.0355-j1.9585	5.6763+j2.0788	6.0450	7hz
8	-0.4	0.6	1.1	-1.1	-2.2	4.4		
9	0.5	0.5	0.1	0.1+j1.6	0.0293+j4.3577	5.6761-j2.0788		
10	-0.4	-0.3	-2.2	3.3	3.3 +j2.9	-0.8828-j3.3840		
11	-0.5	-1.9	1.6	0.1-j1.6	0.1707+j1.1577	-2.0312-j2.8423		
12	-0.9	1.4	0.9	-1.1	0.0	-2.2		
13	-1.9	-0.5	1.9	1.9+j2.0	0.1707-j1.1577	-4.0398+j1.6407		
14	1.6	0.0	-2.0	2.9	3.3 -j2.9	-0.3172-j5.3840		
15	-2.0	-2.0	2.0	1.9-j2.0	0.0293-j4.3577	2.3949-j5.9958		

FFT Algorithm
Figure 1-5

Modified - FFT

Level Coeff	(4N)	(8)	(8N)	(16)	|16|	
0	0.0	2.2	2.2	0.0	0.0	D.C.
1	0.2316+j0.9728	1.2308+j0.9326	1.2308+j0.9326	0.5002+j2.7925	2.8369	1hz
2	-0.6	-0.6 +j1.0	-0.5145+j0.8575	-0.4501+j1.8554	1.9092	2hz
3	0.2316-j0.9728	-0.7676-j1.0128	-0.7676-j1.0128	-0.7353-j0.9308	1.1874	3hz
4	2.2	-2.2	-2.2	-2.2	1.0	4hz
5	0.6783-j0.7348	-0.7676+j1.0128	-0.7676+j1.0128	-0.7999+j1.0948	1.3559	5hz
6	1.0	-0.6 -j1.0	-0.5145-j0.8575	-0.5789+j0.1404	0.5957	6hz
7	0.6783+j0.7348	1.2308-j0.9328	1.2308-j0.9328	1.9614+j0.9269	2.1694	7hz
8	-1.1	-2.2	-2.2	-4.4		
9	0.0624+j0.9981	0.0367+j1.9977	0.0367+j1.9977	1.9614-j0.9269		
10	3.3	3.3 +j2.9	0.7512+j0.6601	-0.5789-j0.1404		
11	0.0624-j0.9981	0.0881+j0.0015	0.0881+j0.0015	-0.7999-j1.0948		
12	-1.1	0.0	0.0	-2.2		
13	0.6887+j0.7250	0.0881-j0.0015	0.0881-j0.0015	-0.7353+j0.9308		
14	2.9	3.3 -j2.9	0.7512-j0.6601	-0.4501-j1.8554		
15	0.6887-j0.7250	0.0367-j1.9977	0.0367-j1.9977	0.5002-j2.7925		

Modified FFT Algorithm
Figure 1-6

	FFT	FFT Normalized to mean of 1.0
D.C	0.0	0.0
f_0	6.4564	1.4371
$2f_0$	5.3933	1.2005
$3f_0$	4.3603	0.9705
$4f_0$	2.2	0.4897
$5f_0$	3.4935	0.7776
$6f_0$	3.4973	0.7785
$7f_0$	6.0450	1.3455

	FFT - Mod	FFT - Mod Normalized to mean of 1.0
D.C.	0.0	0.0
f_0	2.8369	1.5031
$2f_0$	1.9192	1.5243
$3f_0$	1.1874	0.6291
$4f_0$	1.0	1.0
$5f_0$	1.3559	0.7184
$6f_0$	0.5957	0.4756
$7f_0$	2.1694	1.1494

Comparison of the output coefficients of the FFT and Modified FFT before and after they have been given a mean value of 1

Figure 1-7

shows the modified FFT algorithm starting from the 4 level (the FFT and modified FFT algorithms are the same up through the 4 level). Figure 1-7 compares the output coefficients of the modified FFT and the FFT before and after the coefficients have been given a mean value of 1. The FFT coefficients have also been given a mean value of 1 for comparison purposes which shows the value of this procedure for comparing sequences of numbers when there is no known relationship between the two sequences. In the case of the FFT the mean value of the coefficients was computed using all of the coefficients as opposed to the manner in which it was performed in the case of the modified FFT. There is no loss in generality in doing this because dividing a sequence by its mean value in no way alters the relative magnitudes of the coefficients.

It is interesting to note in the case of the modified FFT that the signal frequence f_0 (the 1 hertz sinusoid) appears much larger than $2f_0$ before the "mean 1" process was applied. After applying this process the magnitude of f_0 and $2f_0$ are very close, as expected. Grouping the output coefficients of the modified FFT that depend on normalization at each level and then comparing them with the FFT coefficients it is seen that the signals of the modified FFT have greater magnitude than the signals of the FFT and the noise coefficients are smaller for the modified FFT than for the FFT in every case except $4f_0$.

	FFT		Mod FFT		FFT		Mod FFT
f_0	1.4371	<	1.5031	$2f_0$	1.2005	<	1.5243
$3f_0$	.9705	>	.6291	$6f_0$	.7785	>	.4756
$5f_0$	.7776	>	.7184				
$7f_0$	1.3455	>	1.1494	$4f_0$	.4897	<	1.0

The FFT spectrum was given a mean of 1 by dividing each coefficient by the mean of all of the coefficients. In the case of the modified FFT each coefficient was divided by the mean of the coefficients that were normalized at its normalization level. Each procedure resulted in a spectrum with a mean of 1 but by different methods. To alleviate any doubt that the different methods used to give each process a mean value of 1 were responsible for the apparent improved process gain of the modified FFT over the FFT in this example the following compares the processes where each output spectrum has been given a mean value of 1 by the method used for the modified FFT.

	FFT		Mod FFT		FFT		Mod FFT
f_0	1.2667	<	1.5031	$2f_0$	1.2133	<	1.5243
$3f_0$	.8619	>	.6291	$6f_0$	.7867	>	.4756
$5f_0$	.6854	<	.7184				
$7f_0$	1.1860	>	1.1494	$4f_0$	1.0	=	1.0

The method of normalizing coefficients to a mean of 1 seems to have its merits but still does not solve the problem of giving each frequency the same dependence on phase coherence as opposed to amplitude plus phase coherence in the

case of the standard FFT. Consider the frequency $(N/4)f_0$. It is in fact the FFT coefficient with a magnitude of 1. The frequencies $(N/8)f_0$ and $(3N/8)f_0$ do not have their amplitude influences removed until just prior to the last pass through the algorithm. On the other hand the frequencies nf_0 (n=1,3,5,...,N/2-1) have their amplitude influences removed from their intermediate coefficients early in the transform.

D. METHOD OF IMPLEMENTATION

The method that was finally chosen was that of making all frequencies look like odd harmonics of 1/T. Keeping T constant and using frequency translation was tried but did not solve the problem as the even harmonics of f_0, when translated by 1/T, still behaved like even harmonics of f_0 when normalization was performed at the 4 level. That is, they did not reflect all possible phase angles at the output of the transformation.

The method that did work was to vary T, the length of the transform, so that all frequencies were made to look like odd harmonics of 1/T. This is best explained by example. Consider a time record T seconds long and a sinusoid of frequency 2/T that has been sampled N times in T seconds.

The correlating sinusoids are:

$$\cos(2\pi nk/N)-j\sin(2\pi nk/N).$$

The only non-zero correlation occurs when n=2. Now, if the time record were divided in half, then there would be N/2

samples in time T/2. The correlating sinusoids would then be: $\cos[2\pi nk/(N/2)]-j\sin[2\pi nk/(N/2)]$ and the only non-zero correlation would occur when n=1. The net result is that the correlating sinusoids are identical but the transform length is cut in half and the $2f_0$ hertz sinusoid looks like a sinusoid of frequency f_0 to the time record of length T/2. If two transformations were performed and the output coefficients are summed in the complex domain the net result is the same as if one transformation was performed of length T. This is simply because the correlating sinusoids were identical in both cases, that is, every multiplication was identical in each case as was the summation of the products of these multiplications. If a $4f_0$ sinusoid were made to look like an odd harmonic, then the input record would have to be divided into four equal parts and 4-N/4 point transformations would then have to be summed.

When the normalization modification is then added to the algorithm essentially only one-half of the algorithm is performed, the half that deals with the odd harmonics of f_0. But the algorithm must be performed considerably more times. First all N points are transformed to obtain the frequency coefficients relating to nf_0 (n=1,3,5,...,N-1). Then 2-N/2 point transforms are performed to obtain the frequency coefficients relating to nf_0 (n=2,6,10,...,N-2). Then 4-N/4 point transforms are performed to obtain the frequency coefficients relating to nf_0 (n=4,12,20,...,N-4), etc., until N/4-4 point transforms are performed to obtain the frequency coefficient relating to $(N/4)f_0$.

For a 1024 point transform the time required to obtain all of the frequency coefficients is approximately three times that of a normal FFT if the transformations are performed one after the other. If parallel processing were employed the processing time could be reduced to half of that required for the FFT. If the criterion of $N\text{Log}_2N$ (for the number of complex operations required in a transformation) is used for comparing the relative time required for the two transformations then, for 32 points it requires $32 \log_2(32) = 32\times5 = 160$ for the FFT. For the modified FFT it takes $16\times(5+4+3+2) = 16\times14 = 224$. For a 1024 point transform the FFT requires $1024\times10 = 10,240$ where the modified FFT requires $512\times(10+9+...+2) = 512\times54 = 27,648$. For 8192 points the FFT requires 106,496 and the modified FFT requires 368,640. The numbers do not account for the extra bit reversals, additional ordering of output coefficients, or the normalization procedure required in the modified FFT.

To summarize the results of this transformation: The proposed modification to the FFT algorithm will be more dependent on the phase coherence of a waveform to be transformed and less dependent on the power in the waveform than the FFT. When a sinusoid of the form $A\sin(2\pi nf_0t+\theta)$ $(n=1,2,3,...,N/2-1)$ is transformed the resulting output coefficients will be a constant independent of A, n, or θ. And θ can be recovered from the output coefficient corresponding to n.

The performance of this algorithm has been thoroughly investigated for signals of the form in the preceding paragraph.

Due to the nonlinearity of the process it is difficult to predict how it will perform in general. The next section investigates the performance of the algorithm against that of the FFT under identical conditions for different types of data.

III. PART 2 EXPERIMENTAL RESULTS

A. MEASUREMENTS

In determining the value of the modification to the FFT various types of data were transformed and compared with the FFT transformations. Several measurements were employed to evaluate the relative merits of the FFT modification when compared to the FFT. The output spectrum consisted of the magnitudes of the output coefficients referred to as $\{F_i\}$, $(i=0,1,2,..,N/2-1)$. A coefficient that is related to the frequency of an input signal is referred to as F_s. In general, the objective of signal processing is the identification of a coefficient that is signal related from one that is not signal related. Normally this is accomplished by visual observation of the magnitude of the output spectrums of the transformations. Large coefficients tend to indicate the presence of a signal and just as important, the consistency of the amplitude of an output coefficient from transform to transform indicates the presence of a signal. It is the combination of these two effects that gives reliability to the estimate of signal presence or absence.

What may be one of the best means of determining the presence or absence of a signal is the analysis by an experienced spectrum analyst. Amplitude and consistency of an output coefficient do not always portray all of the information available in a coefficient. There seem to be other characteristics in the behavior of a spectrum from transform

to transform that gives extra support to an analyst's ability to determine if a signal is present or absent. In general it is difficult to explain what these additional characteristics are and it seems that only after a great deal of experience do they figure in to an analyst's logic when he decides if a signal is present or not. The extra signal characteristics generally pertain to a specific type of signal in a particular environment with which the analyst is very familiar; that is, he cannot generalize his experience with a specific class of signals to all types of signals and environments.

The primary methods utilized in determining the merits of the modified FFT were:

1. Estimation of signal enhancement through visual observation.

2. Determining the difference in amplitude between the magnitude of the signal and the magnitude of the mean value of the noise:

$$F_s - \frac{1}{n-1}\{[\sum_{i=1}^{n} F_i] - F_s\} = F_s - \mu_n$$

where μ_n is the mean value of the noise coefficients.

3. Determining the variance of the noise:

$$\sigma_n^2 = \frac{1}{n-1}\{[\sum_{i=1}^{n} (F_i - \mu_n)^2] - (F_s - \mu_n)^2\}$$

normally associating the smaller variance with the better processing gain (referred to as smoothing the noise).

4. Utilization of a threshold detector to determine with what probability F_s would exceed all of the other F_i or all of the F_i within a neighborhood of F_s, such as F_{s-25+i} (i=0,1,2,...,50), where the neighborhood in this case is 25 coefficients on either side of F_s.

5. Comparing:

$$\frac{(F_s - \sigma_n)^2}{2\sigma_n^2}$$

for the different spectrums. The numerator relates how much greater F_s is than the average of the comparison noise coefficients and the denominator relates to the smoothness of the noise coefficients. The form of this equation is identical to that normally used in determining input signal to noise ratios.

Method 5 was used in almost every experiment (it was not used in some of the early experiments) along with one or more of the other methods. It was found that methods 4 and 5 agreed quite closely in almost every case (there were a negligible number of exceptions and then the difference itself was negligible). "Agreed" in the context of the above sentence means that if a transformation was better than another in one test it was also better in the other test. Method 5 is the easiest to compute and relates directly to the signal to noise ratio of the output spectrum. As a result it is used most often in the following pages to make comparisons of the FFT and the modified FFT.

The problem of determining input signal to noise ratios across a particular noise bandwidth was alleviated by comparing the output spectrum of the modified FFT to that of the FFT. In this way the performance of the modification was judged relative to the FFT; i.e., the FFT was used as the standard of comparison.

It was originally hoped that the resulting spectrum from the modified FFT would differ significantly from that of the standard FFT. If that was true, and if the modified FFT had sufficient process gain then the combination of the two spectra could result in a spectrum with a lower σ_n and then hopefully a greater process gain.

Since all of the theoretical work presented so far deals with simple sinusoids at multiples of the fundamental frequency of the transformation a large void is left between what happens in that case and in the case of an input waveform consisting of multiple sinusoids at arbitrary frequencies combined with random noise, Gaussian or not, especially since the process is non-linear.

Mathematical or statistical analysis of the modified FFT algorithm is complicated at best, and since the process is non-linear all conclusions as to its value were developed through experimentation with different types of input data.

B. SINGLE SINUSOID PLUS GAUSSIAN NOISE

The first data chosen for analysis consisted of a simple sinusoid combined with additive Gaussian noise. Most noise generators are designed to generate Gaussian noise and its statistical properties are well known making it the most convenient and generally accepted type of data for comparison purposes. Due to the simplicity of a single sinusoidal signal and the availability of Gaussian noise generators the first experiments were performed on this type of data.

The signal chosen was a 25 hz sinusoid with additive Gaussian noise with a 100 hz bandwidth. The signal plus noise was generated with a "Wavetek" model 132 VCG/Noise Generator. The input waveform was lowpass filtered at 100 hz and sampled at 256 samples per second. The transformations were 2048 points in length (8 seconds in time) with a resulting resolution of 1/8 hz. The noise to signal ratios ranged from 0 to 24 db in 3 db increments, for 3.33 minutes or 25, 8-second transforms at each level.

Figures 2-1 through 2-9 show the results of these transformations. The horizontal axis represents frequency, increasing from left to right. 150 points (output coefficients) are displayed. The center of the highlighted area represents the 25 hz frequency. The left edge is 15.625 hz and the right edge is 34.375 hz. The vertical axis represents the relative magnitude of the coefficients. The fore and aft axis represents time with the most current information being displayed at the front (or bottom) and the oldest information

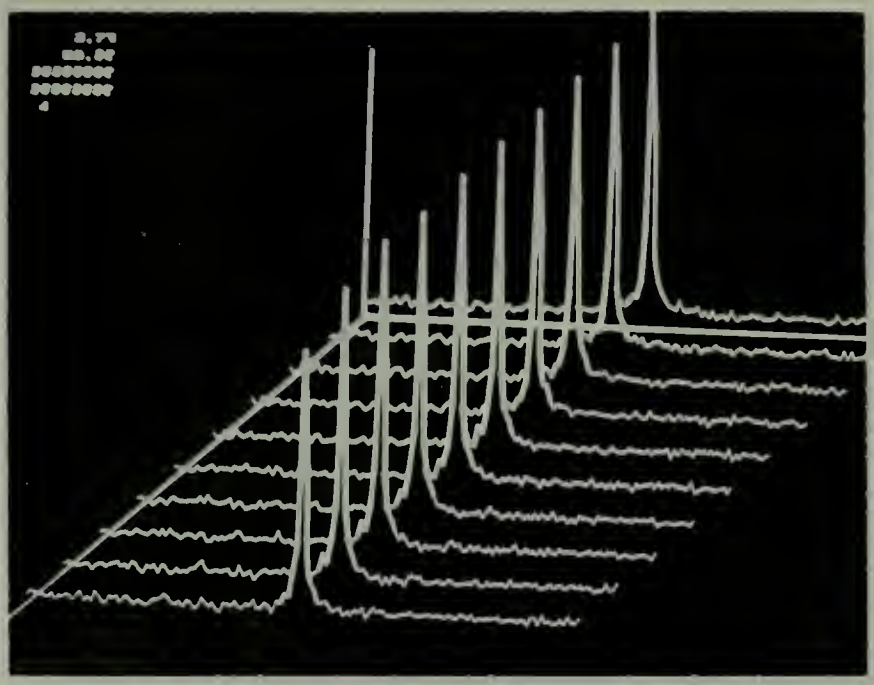
being displayed at the back (or top). Each trace in each figure represents the average of ten transforms or 80 seconds of data. Each trace contains 80% of the same information contained in the traces adjacent to it. A boxcar type averager is used with five compartments. In this case two transforms were averaged in each compartment, then the five compartments were averaged. Each succeeding trace is formed by dumping the compartment with the oldest information (20% of the information in the boxcar) and refilling it with new information, averaging and then displaying. With this in mind then only traces that are separated by five other traces are independent in terms of time from one another.

Table I gives the signal to noise output ratios computed for the average of 25 transforms at each db level. The noise statistics were computed from the 25 coefficients on each side of the signal coefficient. The coefficients immediately to the left and right of the signal were not included in these computations. As stated earlier all S/N output ratios were computed by subtracting the mean of the noise coefficients from the signal coefficient and dividing the square of that result by twice the variance of the noise coefficients.

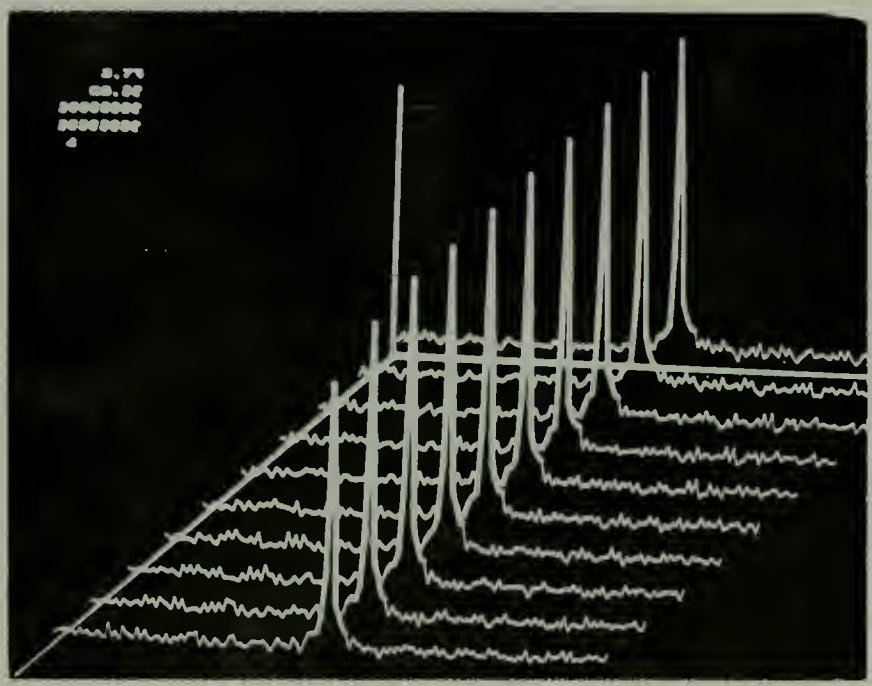
The most obvious result of comparing the respective output spectra of the FFT and modified FFT is their strong similarities. Close observation of the figures reveals the almost identical behavior of the respective spectra as the S/N ratio is reduced. Also the relative magnitude of the

signal coefficient with respect to the magnitude of the coefficients immediately adjacent to the signal coefficient seems to be the same for both spectra. This effect adds another element of similarity, the effect of having the same "shape". The observed variance of the noise appears to be somewhat greater in the modified FFT spectrum.

Table I readily confirms the observed similarities in the spectra. The S/N outputs for the two spectra are very close. The FFT generally has better S/N ratios and this can be attributed to the smaller noise variance observed in the photographs. The FFT becomes significantly better than the modified FFT for the -18, -21, and -24 db S/N ratios, this is again confirmed in the photographs. Comparison of the FFT spectrum squared with that of the FFT spectrum multiplied by the modified FFT spectrum shows no signal enhancement over what can be gained by simply squaring the FFT spectrum.



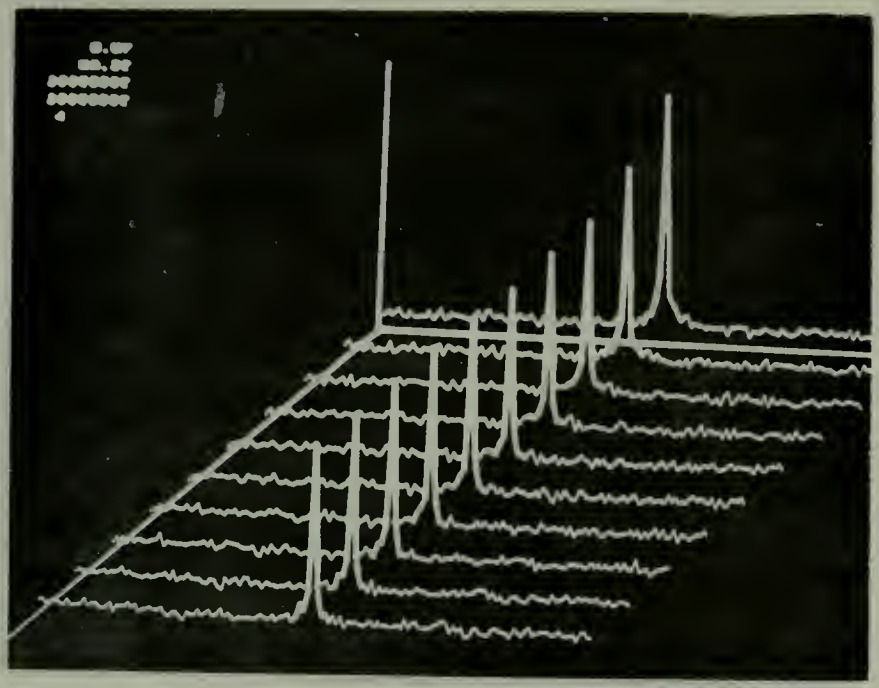
FFT



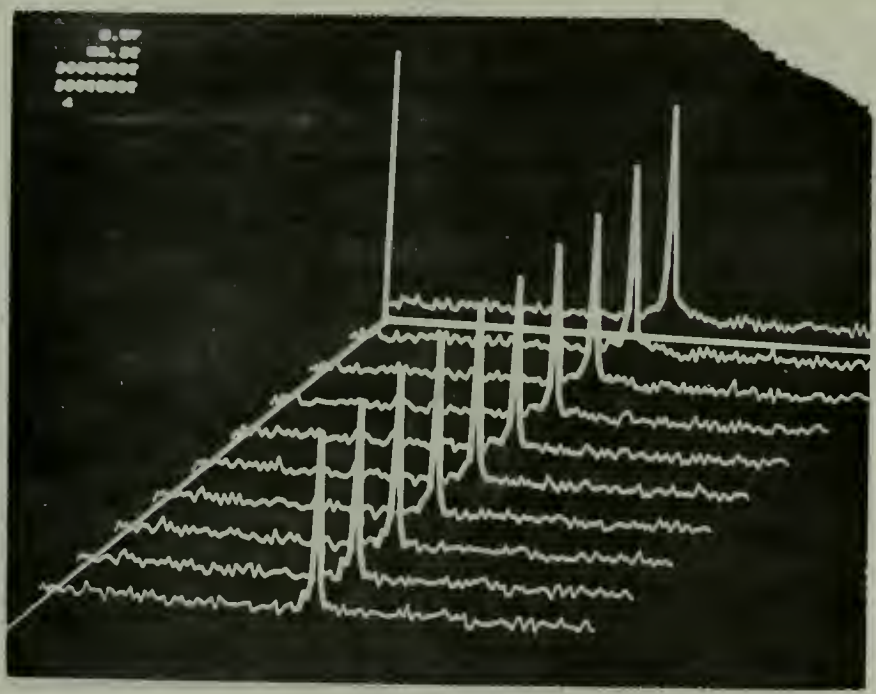
Modified FFT

25 hz signal, S/N = 0 db
10 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

Figure 2-1



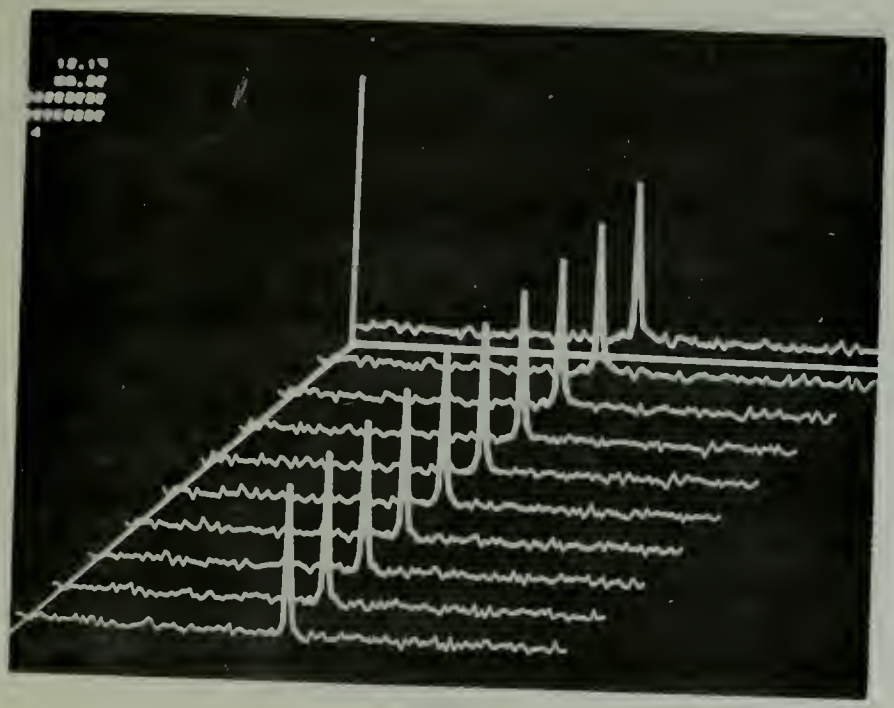
FFT



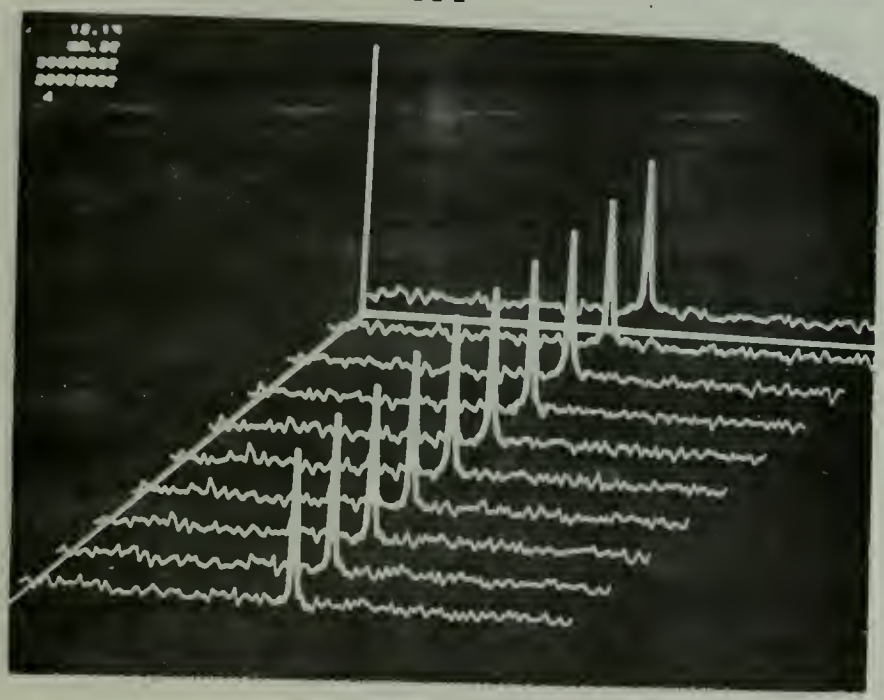
Modified FFT

25 hz signal, S/N = -3 db
10 transformed averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

Figure 2-2



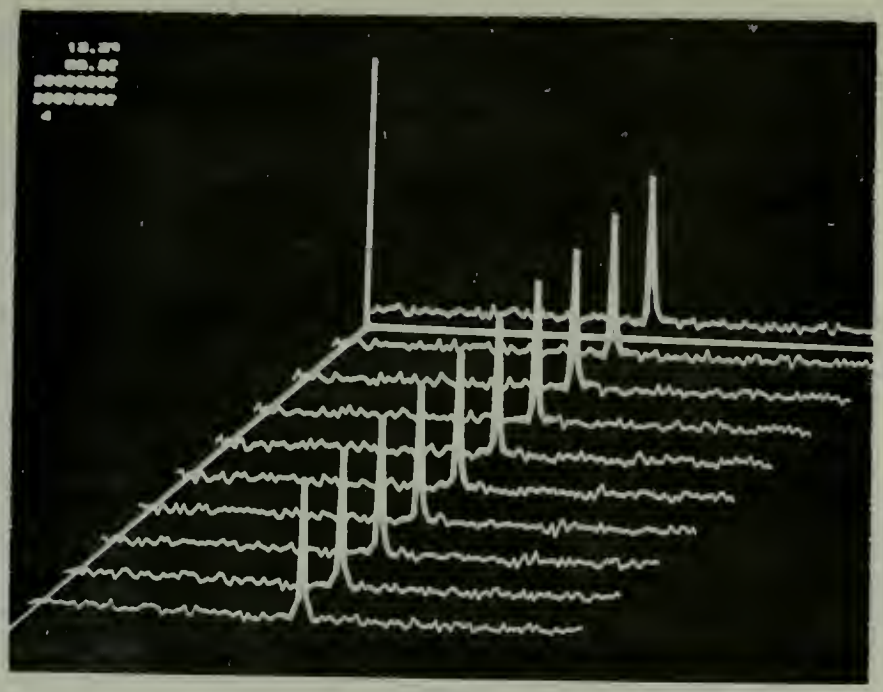
FFT



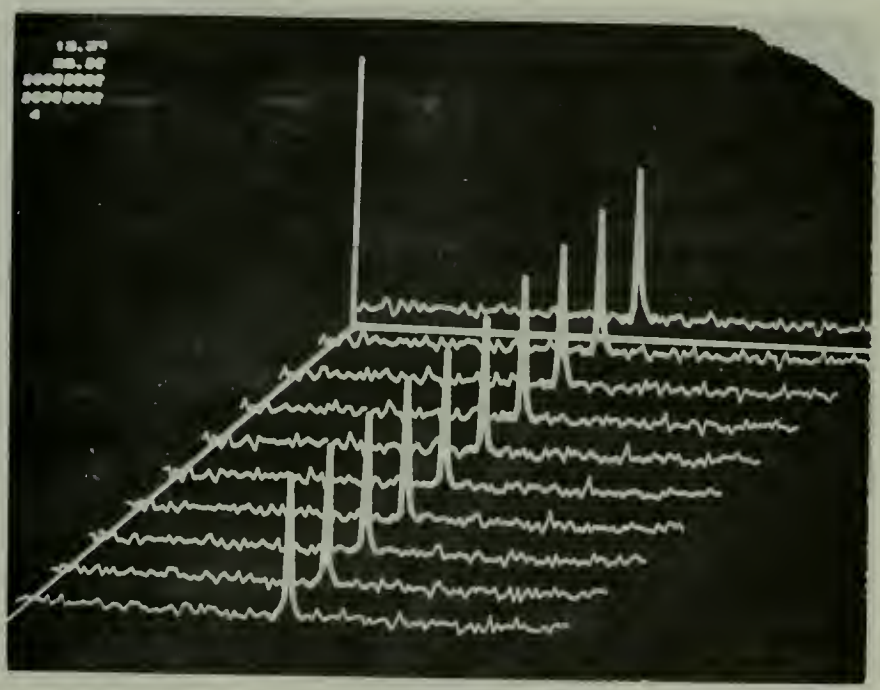
Modified FFT

25 hz signal, S/N = -6 db
10 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

Figure 2-3



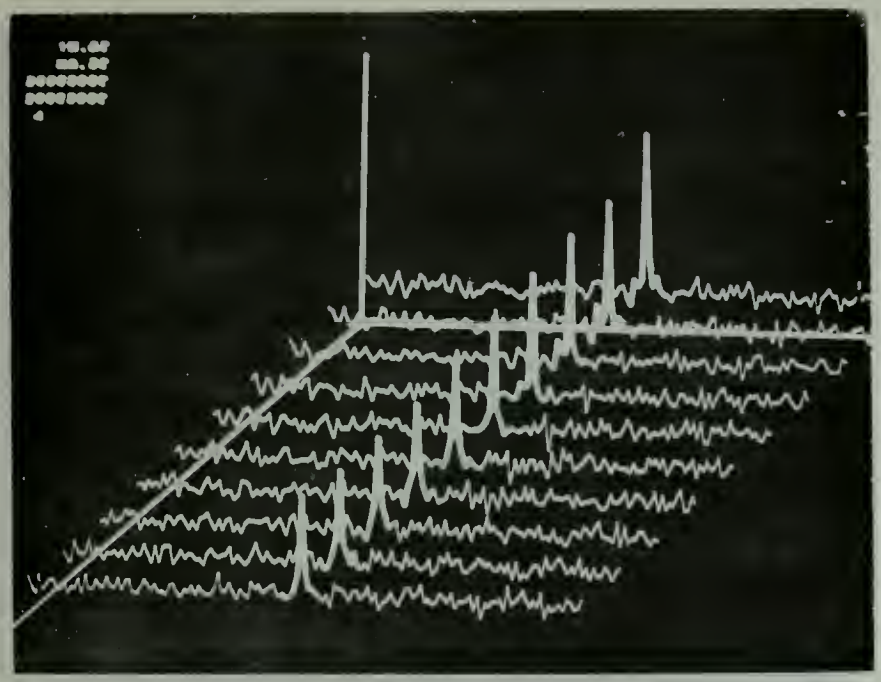
FFT



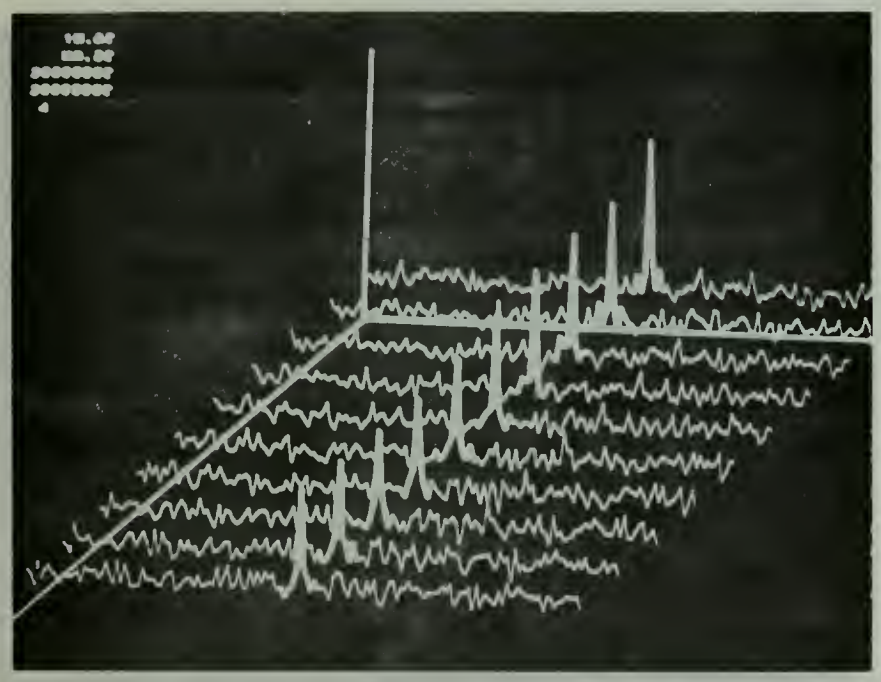
Modified FFT

25 hz signal, S/N = -9 db
10 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

Figure 2-4



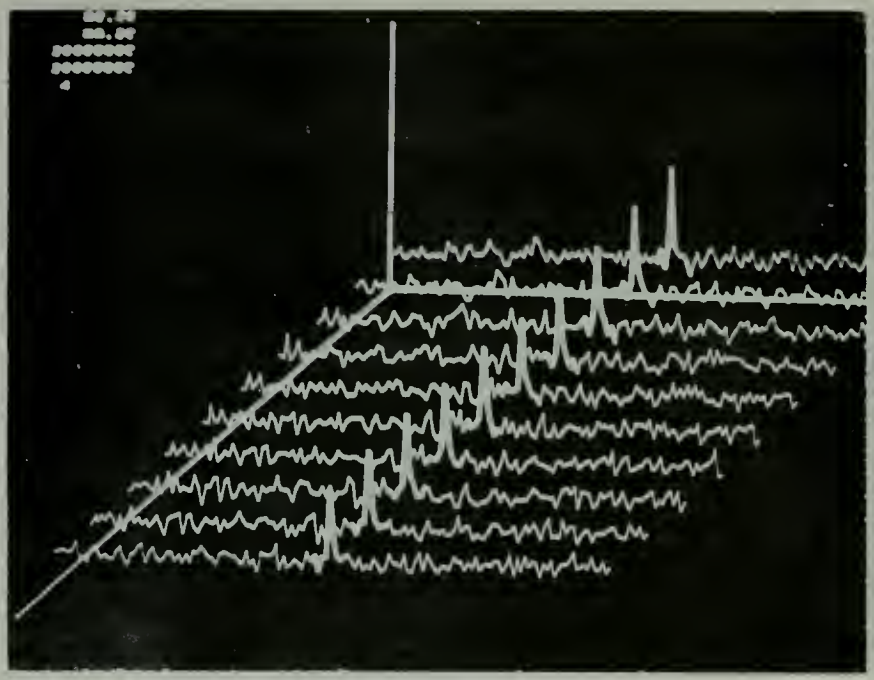
FFT



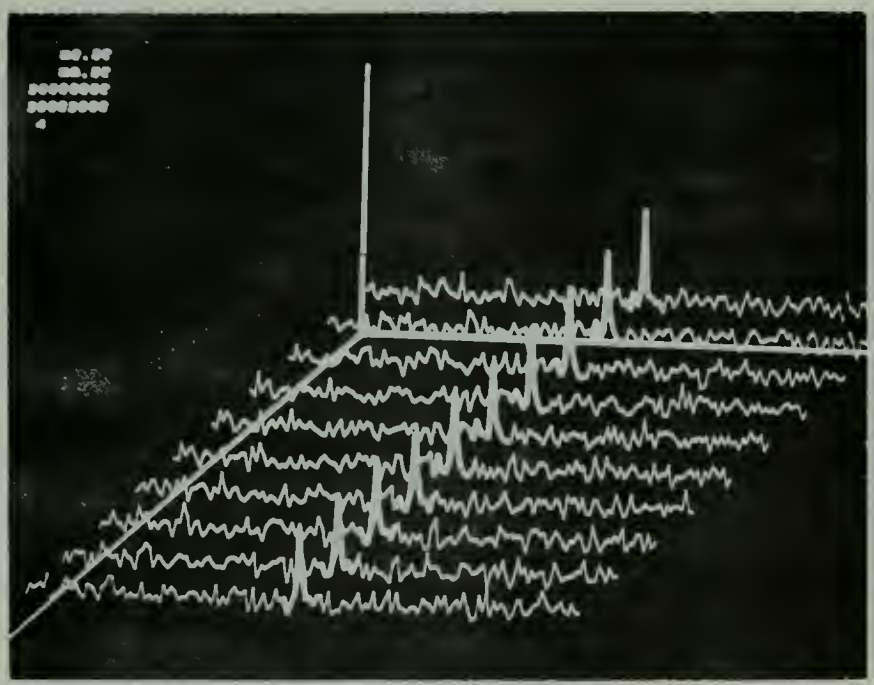
Modified FFT

25 hz signal, S/N = -12 db
10 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

Figure 2-5



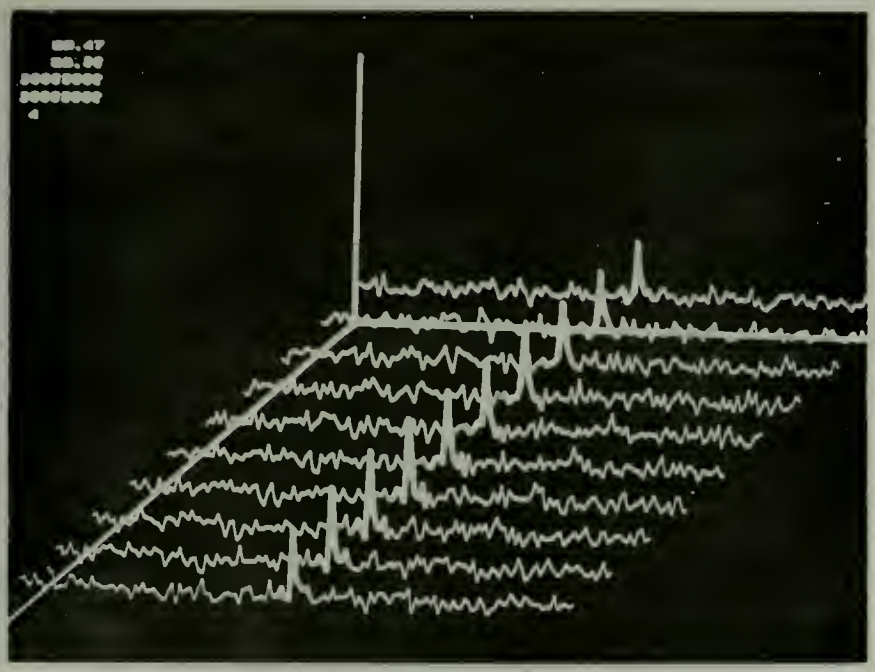
FFT



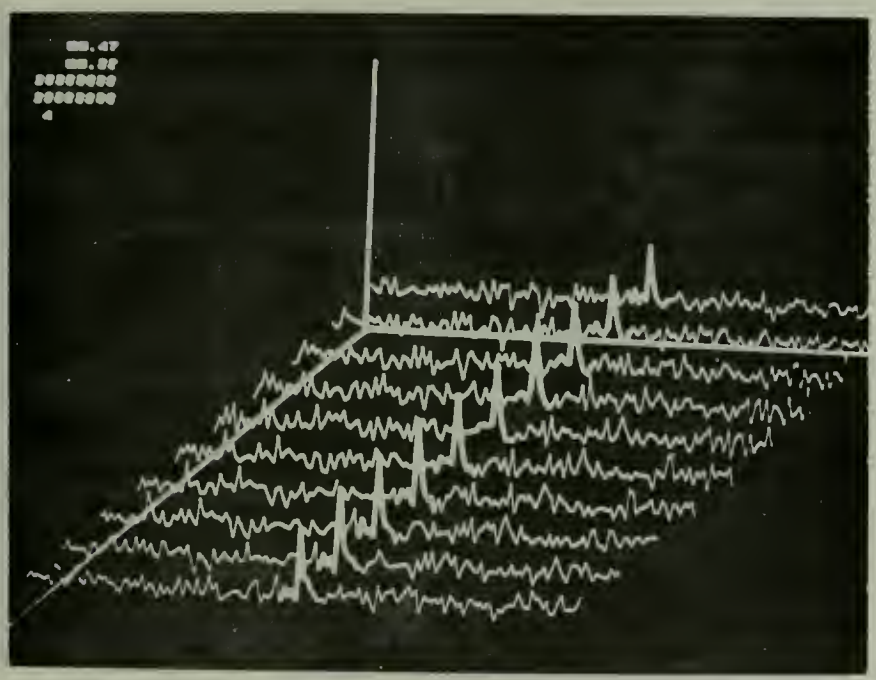
Modified FFT

25 hz signal, S/N = -15 db
10 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

Figure 2-6



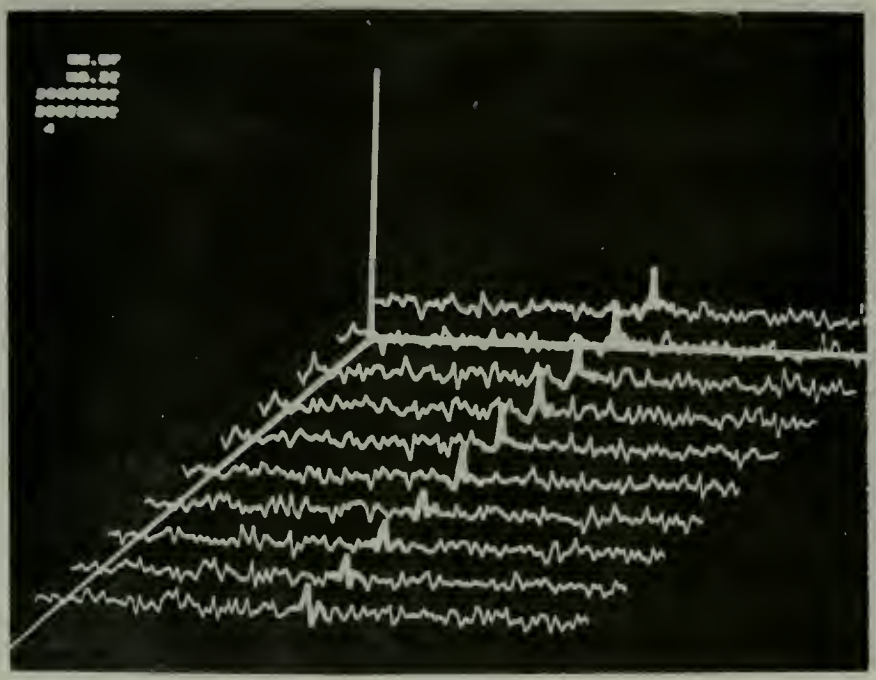
FFT



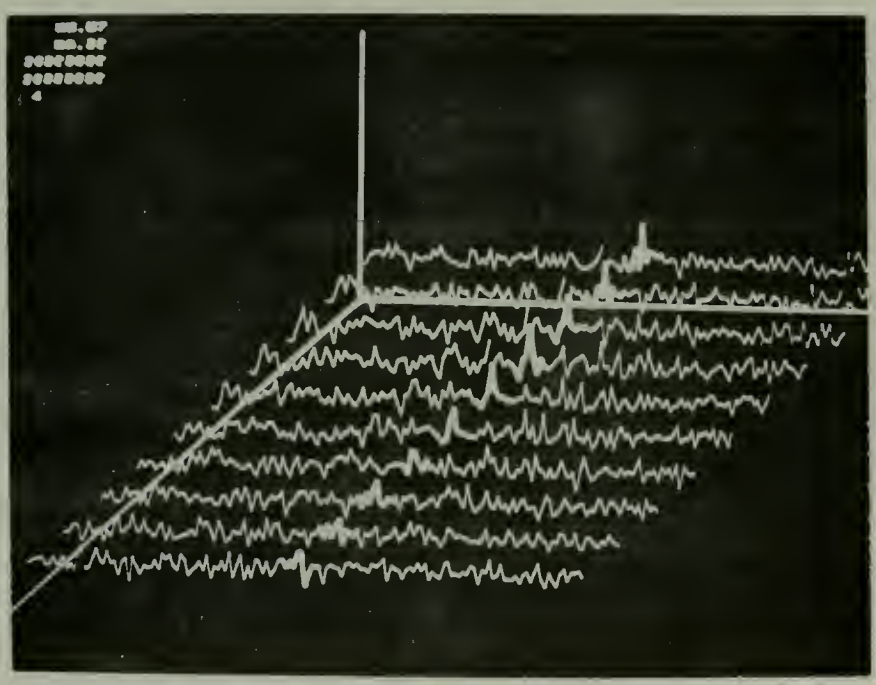
Modified FFT

25 hz signal, S/N = -18 db
10 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

Figure 2-7



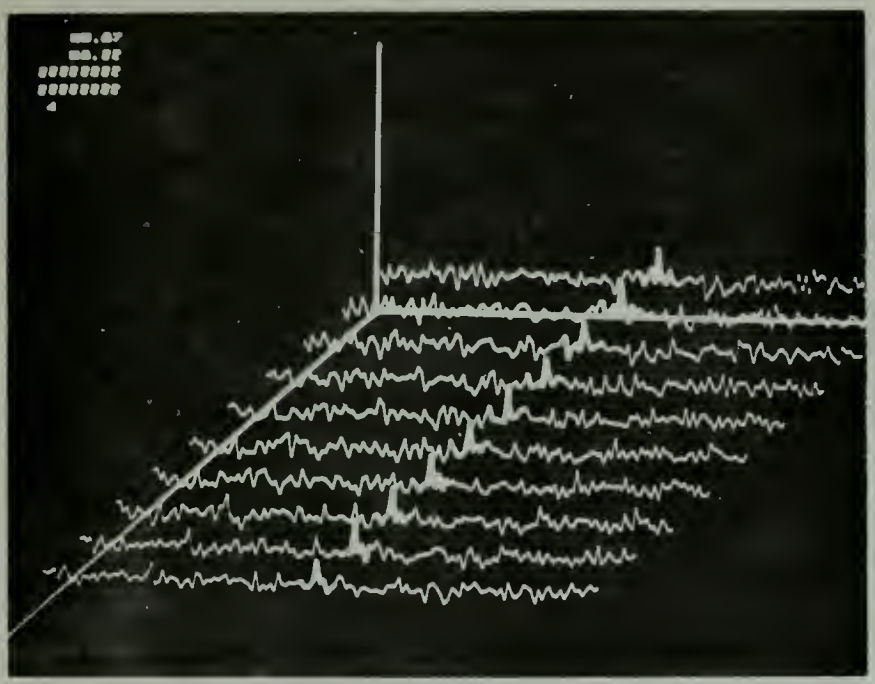
FFT



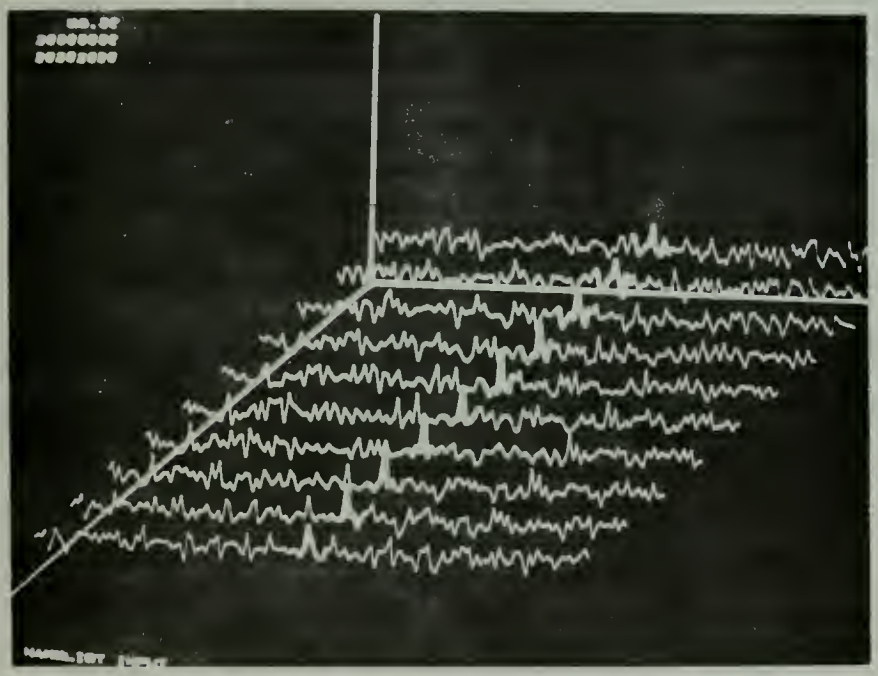
Modified FFT

25 hz signal, S/N = -21 db
10 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

Figure 2-8



FFT



Modified FFT

25 hz signal, S/N = -24 db
10 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

Figure 2-9

S/N in (db	S/N out (db) FFT	Mod-FFT	FFT×FFT	FFT× Mod-FFT
0	26.295	27.260	39.790	40.902
-3	29.734	30.106	42.409	42.414
-6	35.007	32.860	48.469	47.258
-9	34.320	34.173	47.235	47.578
-12	25.480	24.053	32.632	32.208
-15	21.383	20.342	26.925	26.656
-18	21.311	18.646	26.124	25.006
-21	11.519	7.664	13.726	12.490
-24	9.877	2.422	11.457	8.506

Table I

Average S/N out (db) vs. S/N in (db) for the Average of 25 Transforms in the case of the 25 hz signal plus Gaussian noise. S/N out computed as:

$$\frac{(F_s-\mu_n)^2}{2\sigma_n^2}$$

where F_s is the magnitude of the signal coefficient and μ_n and σ_n are computed from 25 noise coefficients on either side of F_s.

C. MULTIPLE SINUSOIDS PLUS GAUSSIAN NOISE

To further investigate the relative merits of the modified FFT Gaussian noise was again used. In this case multiple sinusoidal signals were present allowing investigation into the possibility of destructive interference between the signals due to the normalization procedure.

The data came from a test tape made by the Naval Air Development Center. The resulting spectra are presented in Figures 2-10 through 2-14. The data consisted of four signals in two groups of two. Only one group of two was considered. The tape had ten minutes of data for each signal starting with three minutes of white noise followed by ten minutes of signals at -20 db and -18 db presented in close proximity to each other. These were followed by two more signals for ten minutes at -16 db and -14 db etc., until two signals were presented at 0 db and +2 db (the last two were not transformed). The input signal to noise ratios were measured with respect to a 1 hz noise bandwidth. The figures indicate where the signals are or where they should be.

These figures are interpreted in exactly the same manner as Figures 2-1 through 2-9. In this case forty transforms of 2048 points each were averaged in each trace. The boxcar average was again used. This data was also filtered at 100 hz and sampled at 256 samples per second. The frequency resolution is 1/8 hz.

It should be noted that there was some question as to the reliability of the -18 db signal and it was not considered

to be experimentally acceptable in the data analysis. The indication of where it should be in Figures 2-14 is only a best guess.

Table II gives the average computed output S/N ratios for the summation of 40 transforms. These were computed using the same method employed throughout this study. In this case, when a signal was on the right only 25 coefficients to the right of that signal were considered in computing the noise statistics. Likewise, if the signal was on the left only 25 coefficients to the left of that signal were considered in computing the noise statistics. In all cases the nearest coefficient to the signal coefficient was not considered in the computations.

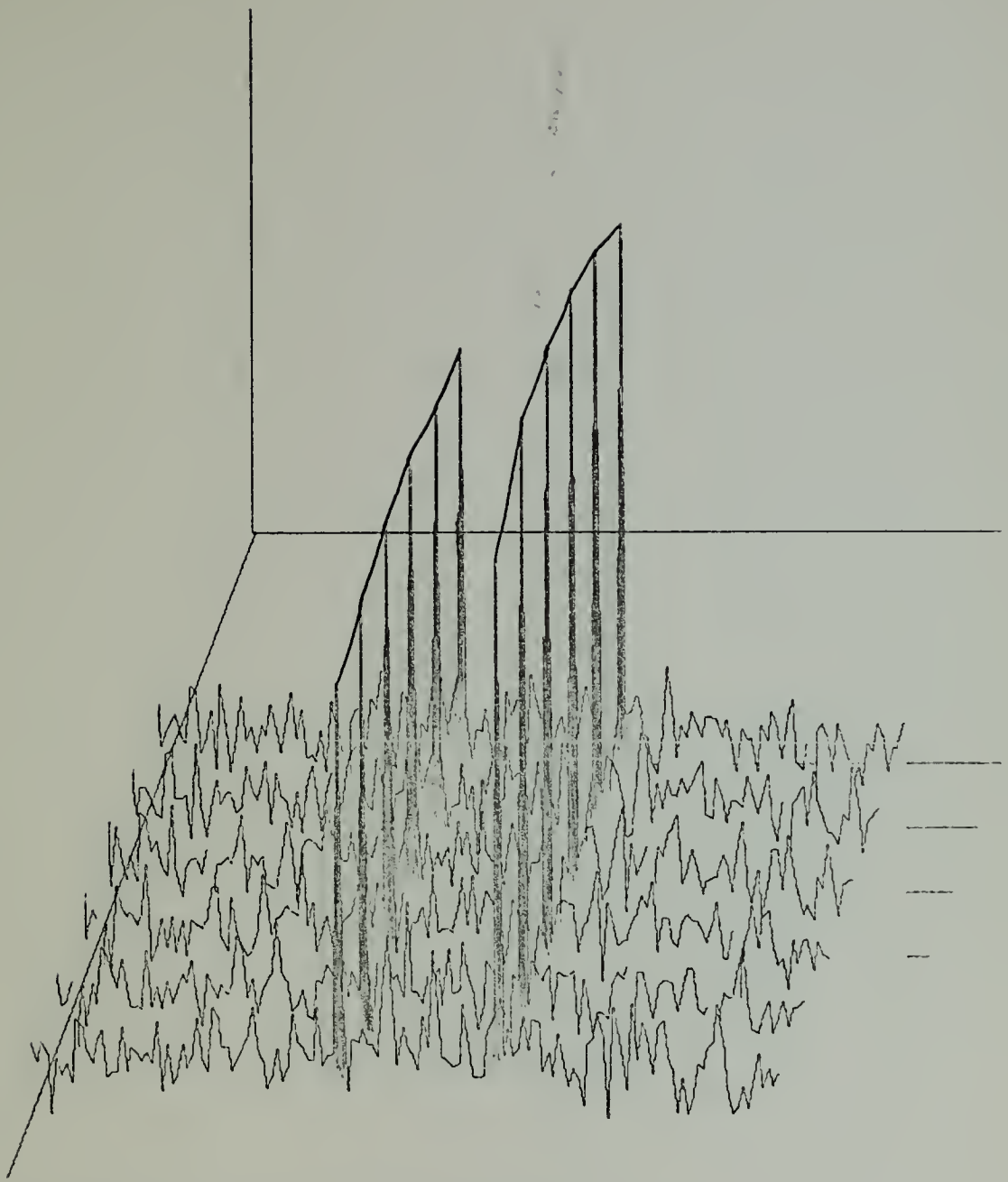
Table III is similar to Table II except that 80 transforms were averaged instead of 40.

The most important result of this experiment was to show that multiple signals did not interfere with each other and that the normalization process did not appear to destroy or reduce signal resolutions. Comparison of the figures again reveals similar spectra for the respective transformations. Comments identical to those made in Section III-B with respect to the single sinusoid plus Gaussian noise case could be made here with one exception. The spectra for the -16 and -20 db signal to noise ratios tend to favor the modified FFT. In Figures 2-13 the top trace of the modified FFT shows the -16 db signal to be nearly as large as the -14 db signal and in the second trace from the top it is actually

larger. Also, the variance of the adjacent noise coefficients appears to be smaller in the modified FFT spectrum than in the FFT spectrum. The individual distinctions are not as pronounced in the case of the -20 db signal in Figures 2-14 but the general appearance tends to favor the modified FFT spectrum.

Table II indicates that the FFT generally outperformed the modified FFT when the signal to noise ratios were relatively small. The reverse was true for the -16 and -20 db S/N ratios (as indicated above the -18 db signal is not being considered in the analysis). As in the case of the single sinusoid presented in Section III-B no obvious trends were present in this data. Again the product of the two spectra when compared against the FFT spectrum squared did not prove to have any particular advantage.

Table III compares the measured results when 80 transforms were averaged instead of 40 as in Table II. The improvement in S/N output was relatively consistent for all S/N ratios and for both spectra with neither spectrum gaining any particular advantage over the other as a result of the deeper average.

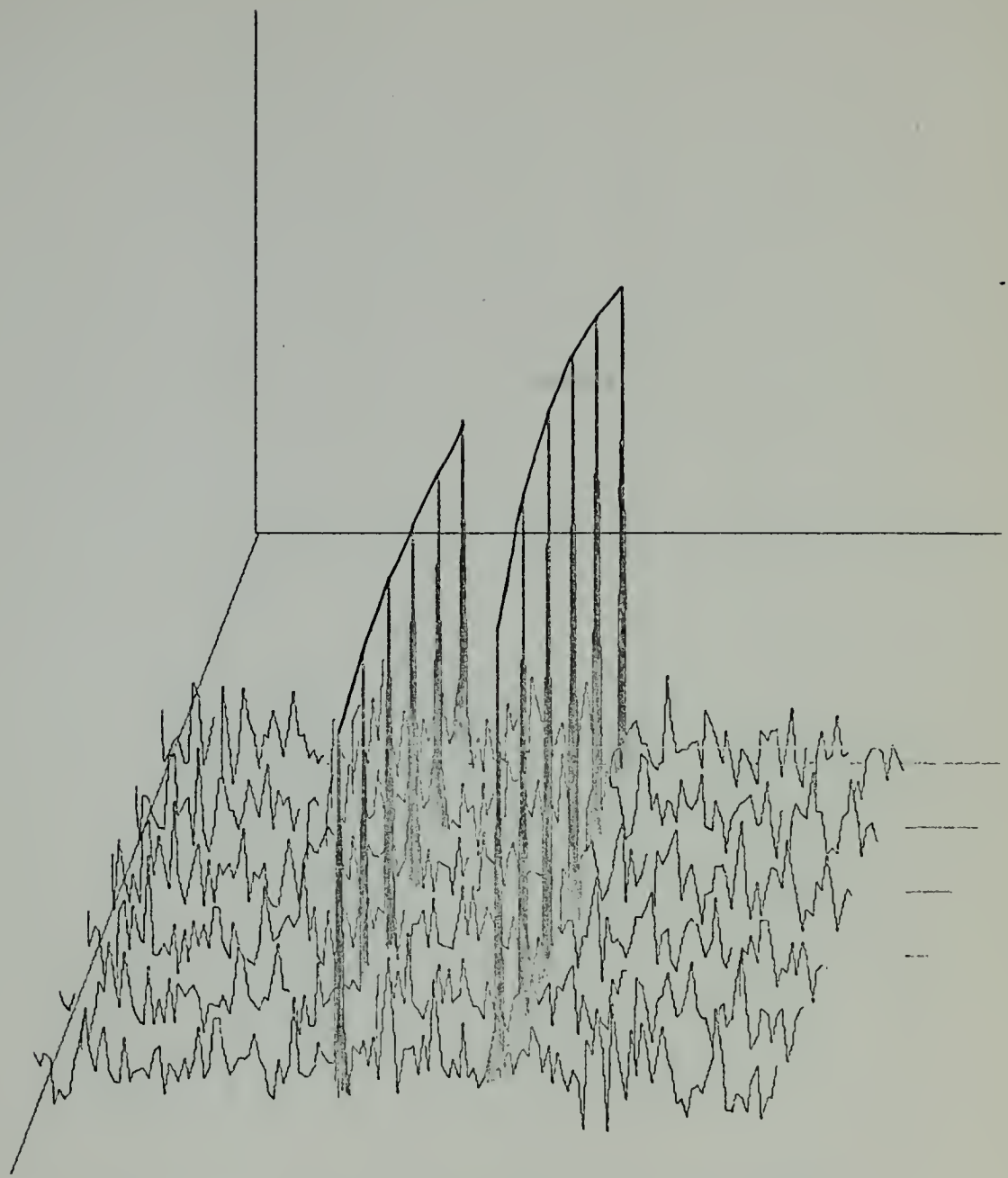


40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution
-2 db signal on right, -4 db signal on left

X-SCALE - 3.88E-01 UNITS/INCH.
Y-SCALE - 3.98E-01 UNITS/INCH.

NADC TEST TAPE (FFT)
S/N=-2,-4 DB

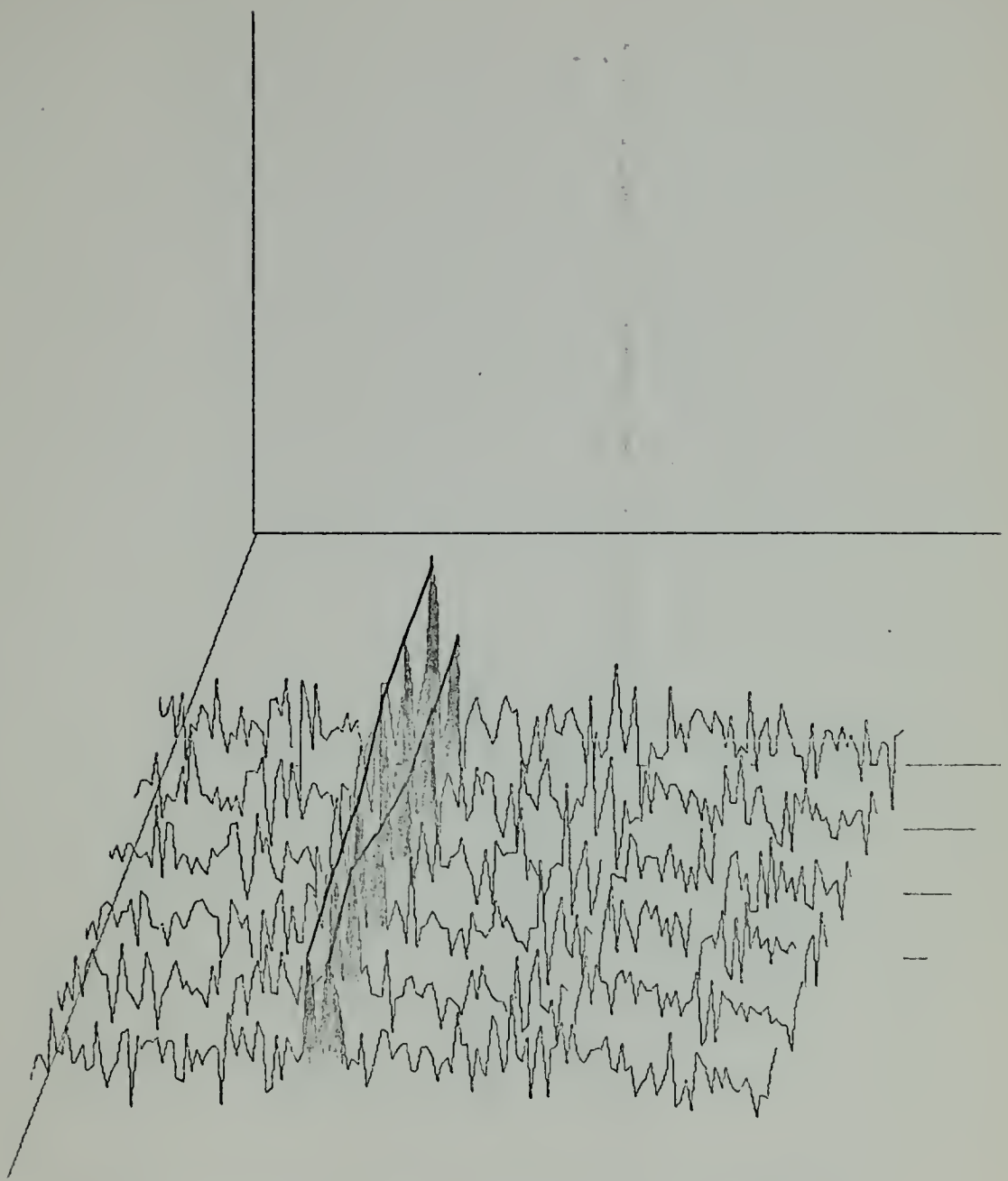
Figure 2-10 (FFT)



40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution
-2 db signal on right, -4 db signal on left

X-SCALE = 0.00E-01 UNITS/INCH.
Y-SCALE = 0.00E-01 UNITS/INCH.
NADC TEST TAPE (MOD FFT)
S/N= -2, -4 DB

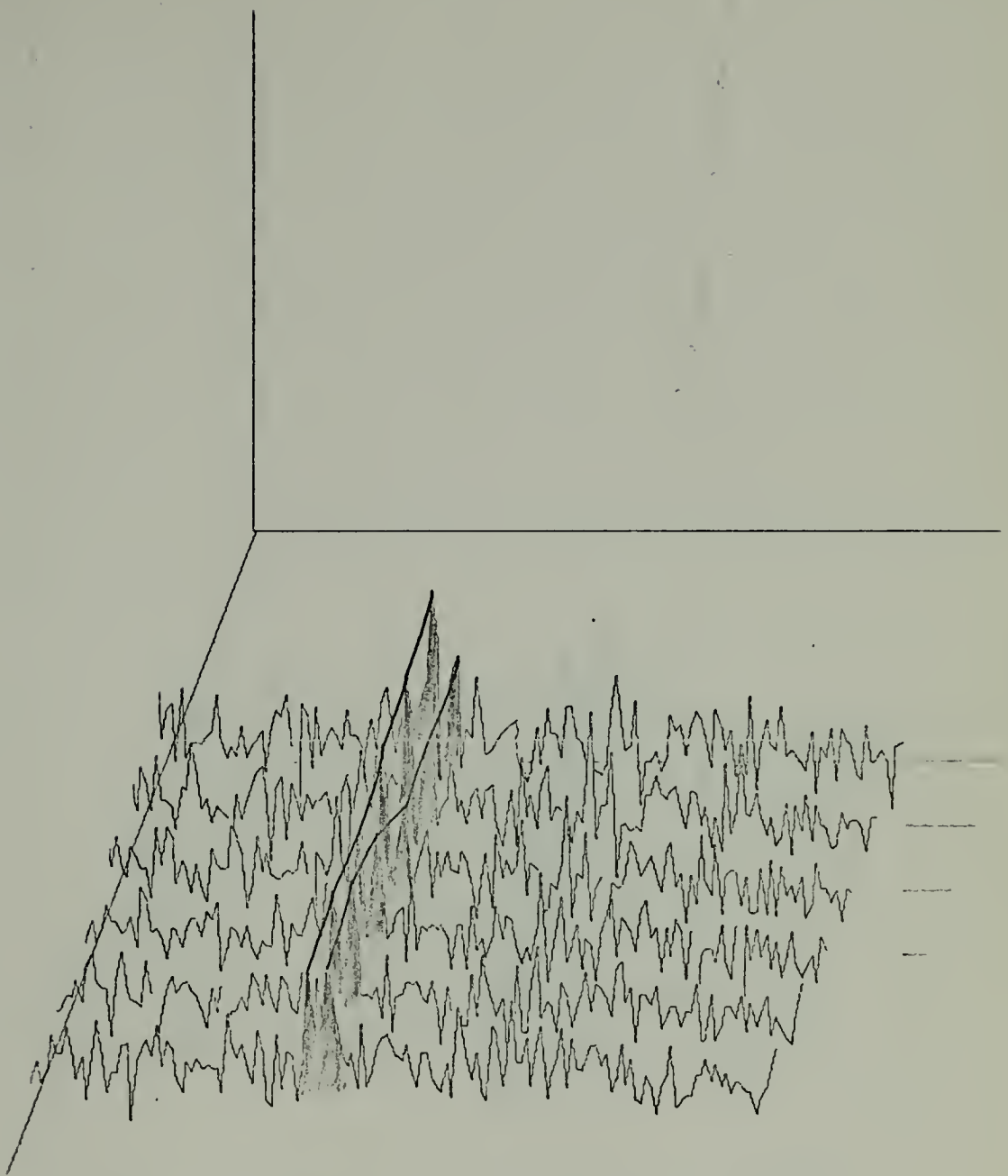
Figure 2-10 (Mod-FFT)



40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution
-6 db signal on left, -8 db signal on right

X-SCALE = 1.00E-01 UNITS/INCH.
Y-SCALE = 1.00E-01 UNITS/INCH.
NADC TEST TAPE (FFT)
S/N= -6, -8 DB

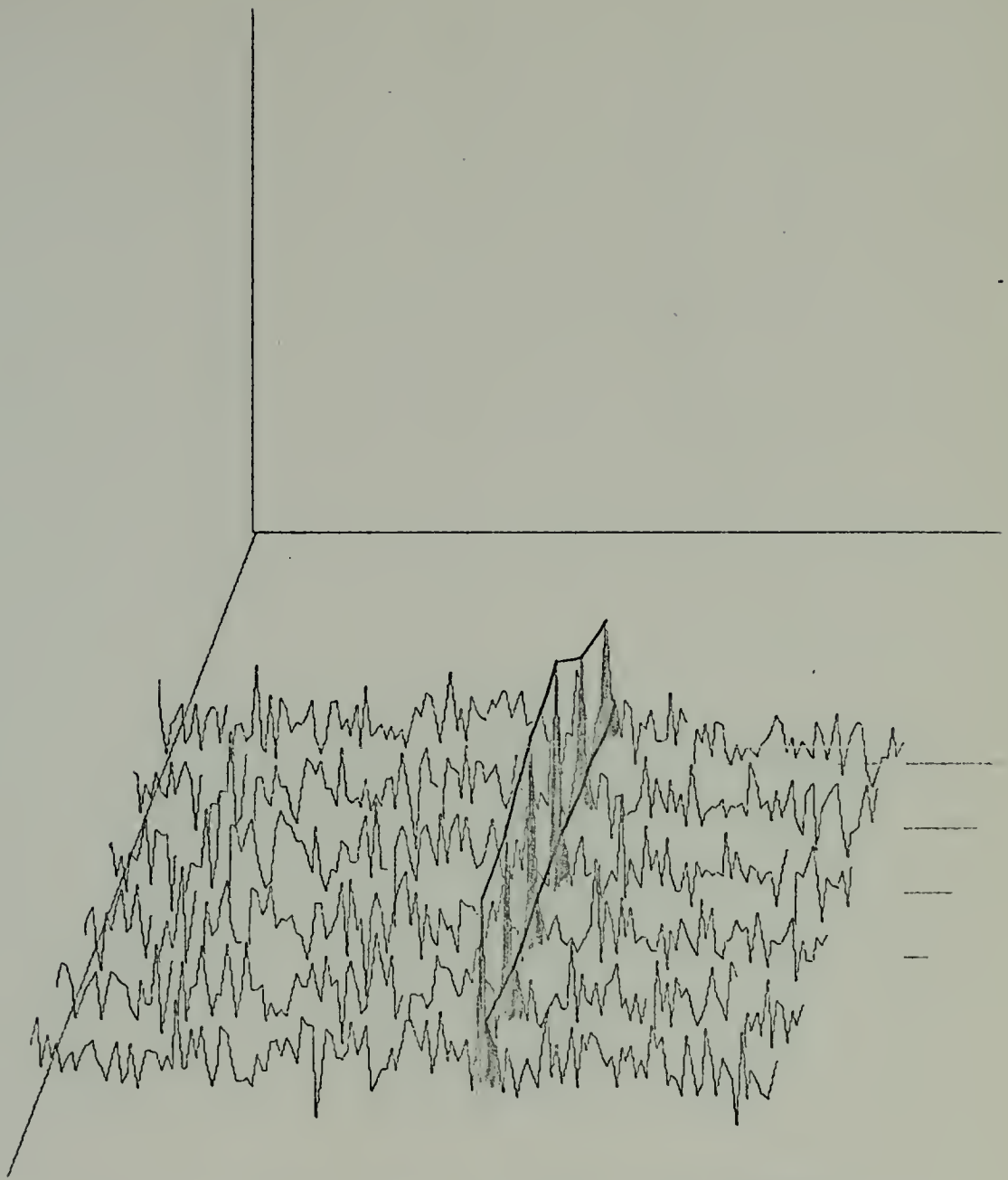
Figure 2-11 (FFT)



40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution
-6 db signal on left, -8 db signal on right

X-SCALE = J.00E-01 UNITS/INCH.
Y-SCALE = J.00E-01 UNITS/INCH.
NADC TEST TAPE (MOD FFT)
S/N= -6, -8 DB

Figure 2-11 (Mod-FFT)

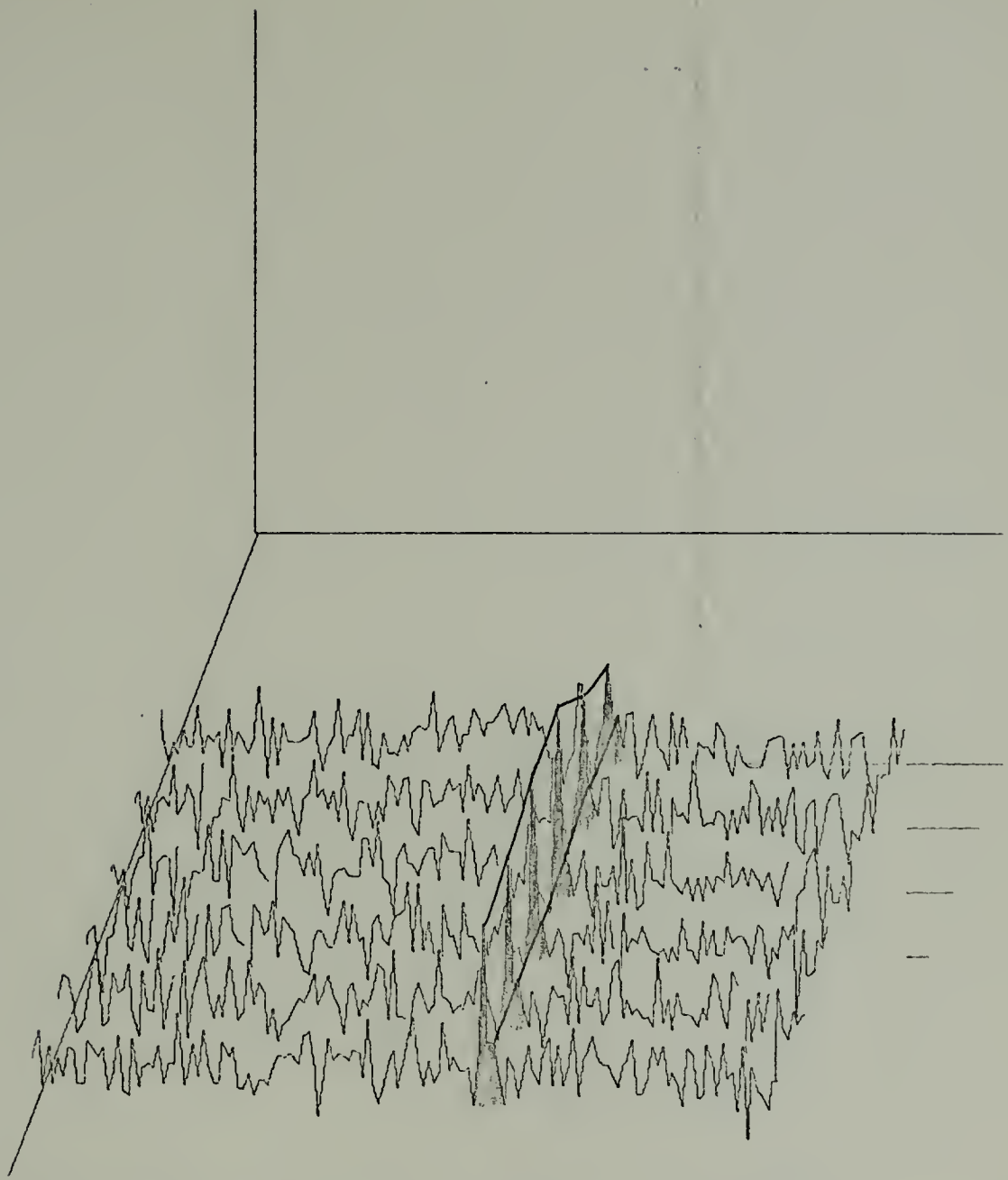


40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution
-10 db signal on left, -12 db signal on right

X-SCALE = 3.00E-01 UNITS/INCH.
Y-SCALE = 3.00E-01 UNITS/INCH.

NADC TEST TAPE (FFT)
S/N= -10, -12 DB

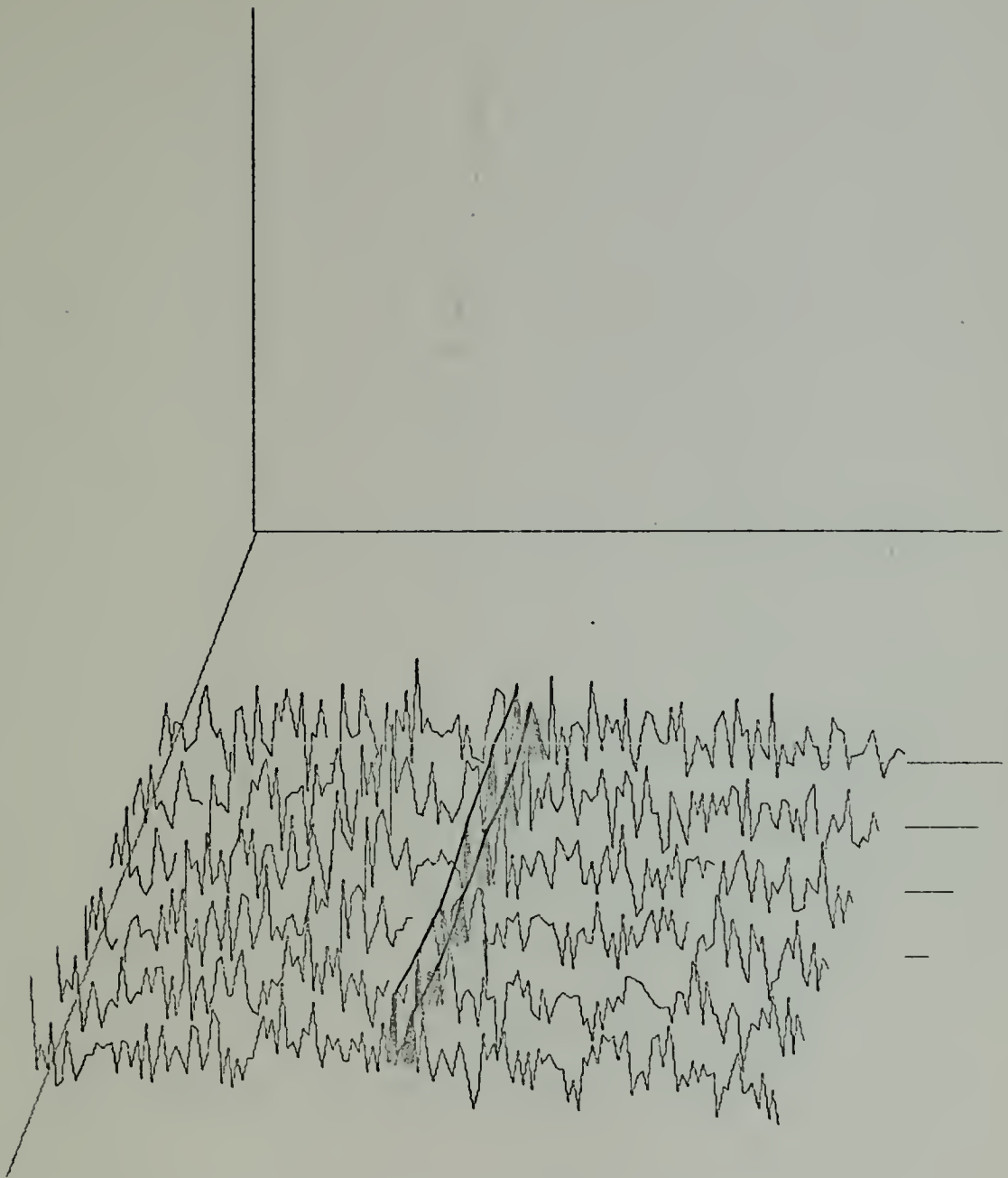
Figure 2-12 (FFT)



40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution
-10 db signal on left, -12 db signal on right

X-SCALE = 3.00E-01 UNITS/INCH.
Y-SCALE = 3.00E-01 UNITS/INCH.
NADC TEST TAPE (MOD FFT).
S/N=-10,-12 DB

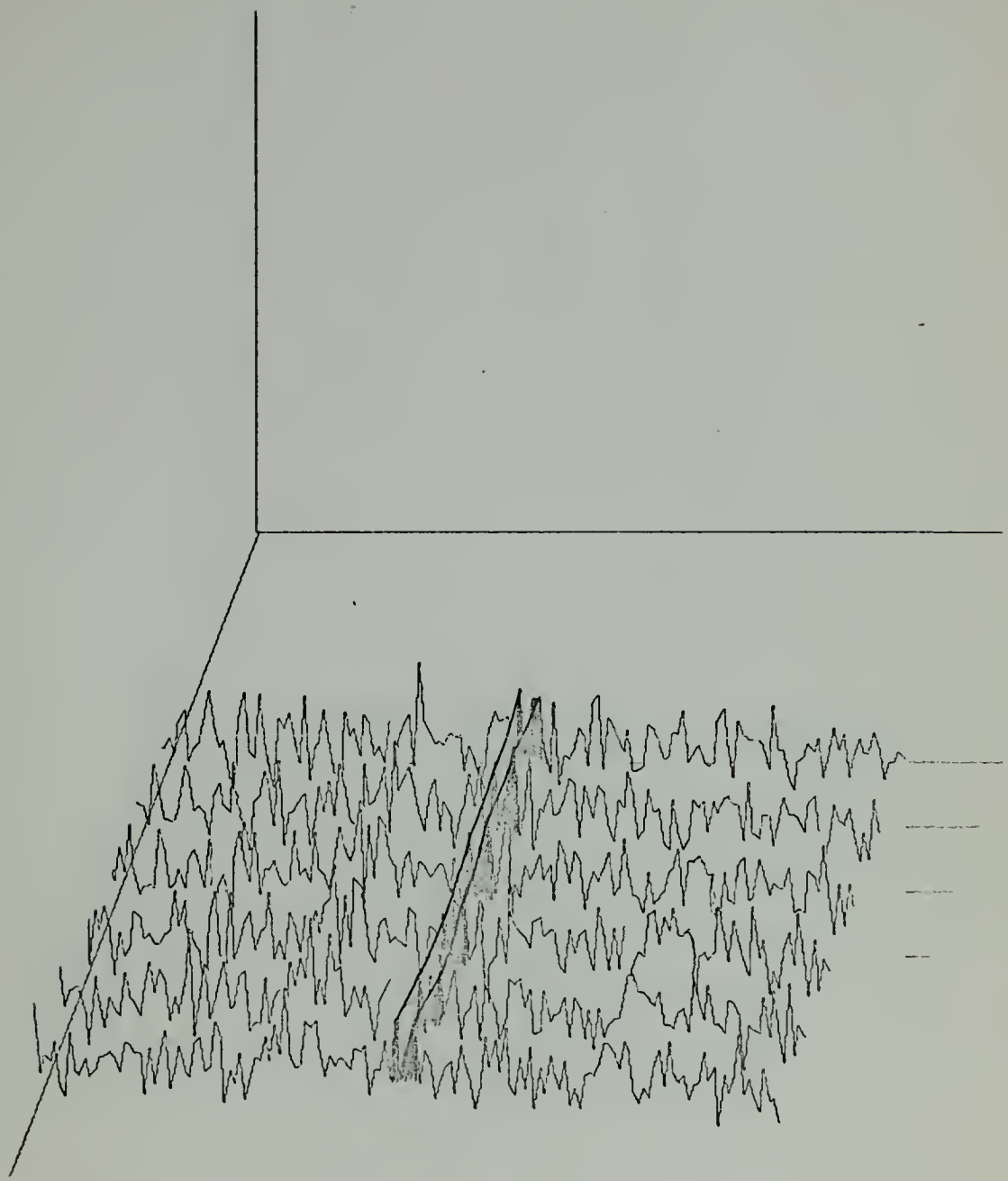
Figure 2-12 (Mod-FFT)



40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution
-14 db signal on left, -16 db signal on right

X-SCALE = 3.00E-01 UNITS/INCH.
Y-SCALE = 3.00E-01 UNITS/INCH.
NADC TEST TAPE (FFT)
S/N= -14, -16 DB

Figure 2-13 (FFT)

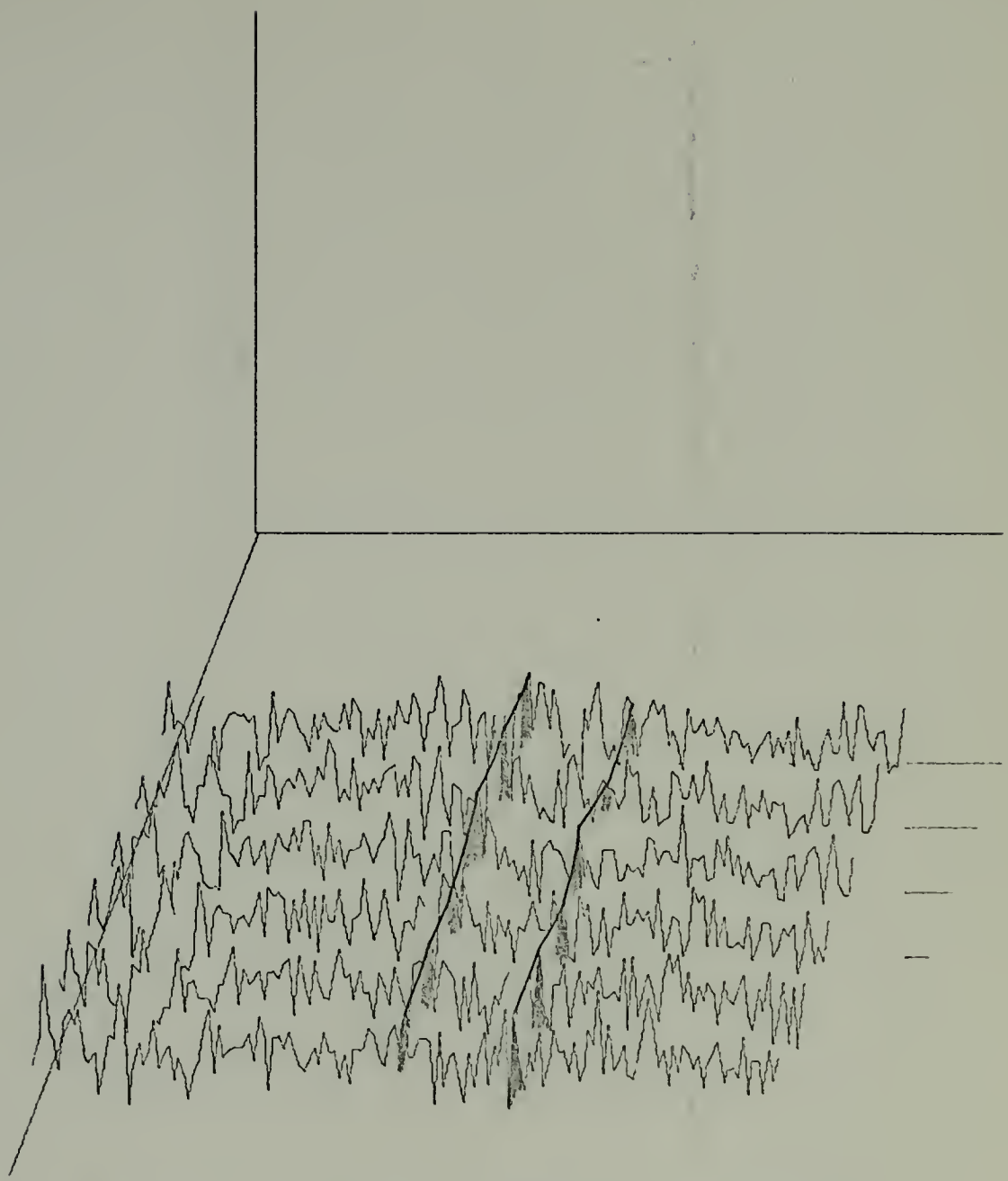


40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution
-14 db signal on left, -16 db signal on right

X-SCALE = 1.00E-01 UNITS/INCH.
Y-SCALE = 1.00E-01 UNITS/INCH.

NADC TEST TAPE (MOD FFT)
S/N=-14,-16 DB

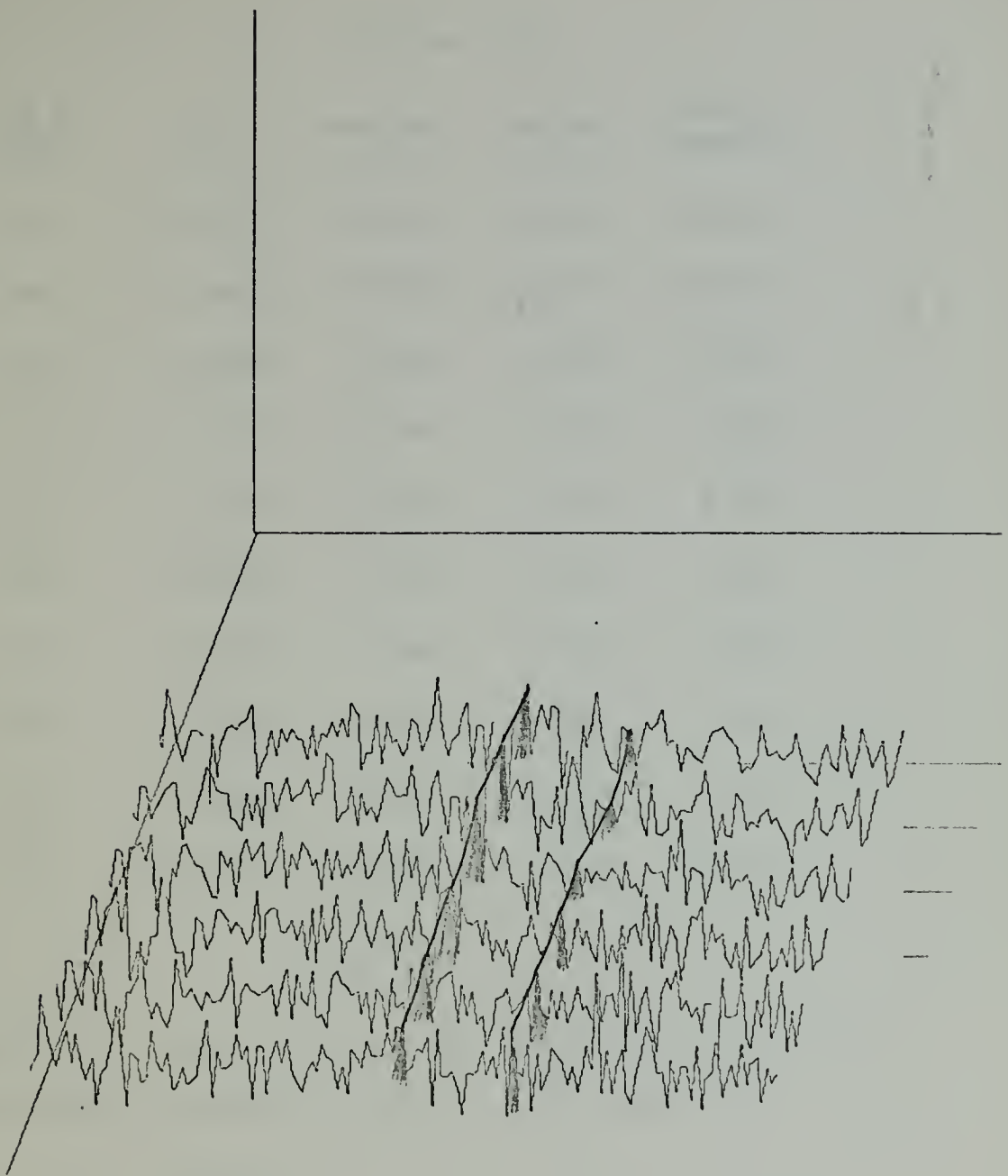
Figure 2-13 (Mod-FFT)



40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution
-18 db signal on right, -20 db signal on left

X-SCALE - 3.88E-01 UNITS/INCH.
Y-SCALE - 3.88E-01 UNITS/INCH.
NADC TEST TAPE (FFT)
S/N=-18,-20 DB

Figure 2-14 (FFT)



40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution
-18 db signal on right, -20 db signal on left

X-SCALE = 3.00E-01 UNITS/INCH.
Y-SCALE = 3.00E-01 UNITS/INCH.
ADC TEST TAPE (MOD FFT)
/N= -18, -20 DB

Figure 2-14 (Mod-FFT)

S/N in (db)	S/N out (db)			
	FFT	Mod-FFT	FFT×FFT	FFT× Mod-FFT
-2	19.655	18.059	23.985	23.165
-4	17.086	16.388	20.218	20.092
-6	2.064	5.086	0.358	2.521
-8	5.901	8.397	4.622	7.018
-10	6.882	5.693	4.769	5.373
-12	8.215	7.121	6.761	6.641
-14	8.847	7.258	7.499	7.160
-16	5.489	6.386	4.008	5.015
-18	5.399	8.119	3.086	5.394
-20	4.370	6.872	3.546	5.865

Table II

Average S/N out (db) vs. S/N in (db) for the Average of 40 Transforms from the NADC Test Tape. S/N computed as:

$$\frac{(F_s-\mu_n)^2}{2\sigma_n^2}$$

where F_s is the magnitude of the signal coefficient and μ_n and σ_n are computed from 25 noise coefficients on one side of F_s.

S/N in (db)	S/N out (db) FFT	Mod-FFT	FFT×FFT	FFT× Mod-FFT
-2	23.669	22.083	28.316	27.606
-4	19.943	19.738	23.094	23.300
-6	3.334	7.066	1.217	3.991
-8	10.492	10.054	8.871	9.427
-10	8.575	8.366	6.420	7.140
-12	10.395	8.889	8.827	8.437
-14	10.895	10.048	9.414	9.021
-16	7.355	9.487	5.707	7.171
-18	7.504	11.319	4.676	7.294
-20	7.032	10.372	6.065	8.581

Table III

Average S/N out (db) vs. S/N in (db) for the Average of 80 Transforms from the NADC Test Tape. S/N computed as:

$$\frac{(F_s-\mu_n)^2}{2\sigma_n^2}$$

where F_s is the magnitude of the signal coefficient and μ_n and σ_n are computed from 25 noise coefficients on one side of F_s.

D. SINGLE SINUSOID PLUS GAUSSIAN NOISE PLUS BURST NOISE

The data to be investigated in this section is similar to that discussed in Section III-B, except in this case additional noise was added in the form of random bursts spaced at random intervals with random duration. The data was generated in this manner to accentuate the effect of spectrum dominance by vectors with large amplitudes. This type of data, where large bursts of noise could dominate or corrupt the resulting spectrum, provides a good test of the FFT modification. If the modification is performing as anticipated it will reduce the spectrum distortions caused by the large amplitudes of the burst noise by weighting all of the intermediate coefficients equally. Then large output coefficients will only be obtained through phase coherent vector additions and not through vectors with large amplitudes.

In this case the signal was set at 10 hz vice 25 hz and additional noise was added in the form of bursts with random length and spaced at random intervals. Figures 2-15 through 2-17 give samples of the input waveform. In Figure 2-15 the top trace shows the signal plus the burst noise, the bottom figure shows it after it has been filtered at 100 hz. Figures 2-16 and 2-17 show the burst noise alone in the top trace, the signal plus noise plus burst noise in the middle trace and the bottom trace is the middle trace after it has been filtered at 100 hz. In each figure the time scale is 0.25 seconds per division.

As was previously done the data was sampled at 256 samples per second and 2048 point transforms were then computed. Figures 2-18 through 2-21 show the results of the transformations. Each trace contains the average of 40 transforms and as was done previously there is 80% redundancy between adjacent traces. The S/N ratios on the figures refer to the S/N ratios of the signal to continuous noise and do not account for the burst noise.

Figures 2-22 through 2-28 are photographs of the same material. The photographs are presented because the graph routine that drew Figures 2-18 through 2-21 would not function if fewer than five traces were to be drawn. Figures 2-22 through 2-24 present the transforms in the same manner as Figures 2-18 through 2-21. Each trace is the summation or average of 40 transforms. Figures 2-25 through 2-28 present the results of averaging 80 transforms. Figure 2-25 shows the -24 db S/N transforms and Figure 2-26 is the same presentation only in this case the coefficients have been squared. It is felt that squaring a spectrum gives better insight into the signal recognition problem. If the signal is readily discernable before the coefficients are squared, then after squaring it will be even more discernable. If the signal becomes less discernable after squaring, then it is concluded that the S/N ratio is insufficient to allow any degree of signal recognition.

Figures 2-27 and 2-28 present the -30 db transforms exactly as described for the -24 db transforms.

Tables IV and V give the signal to noise output ratios for the different S/N ratios. Table IV is the average S/N output ratio for the summation of 40 transforms, and Table V is for the summation of 80 transforms. In each of these cases the noise statistics were computed using 25 coefficients on either side of the signal coefficient, not including the coefficients adjacent to the signal coefficient.

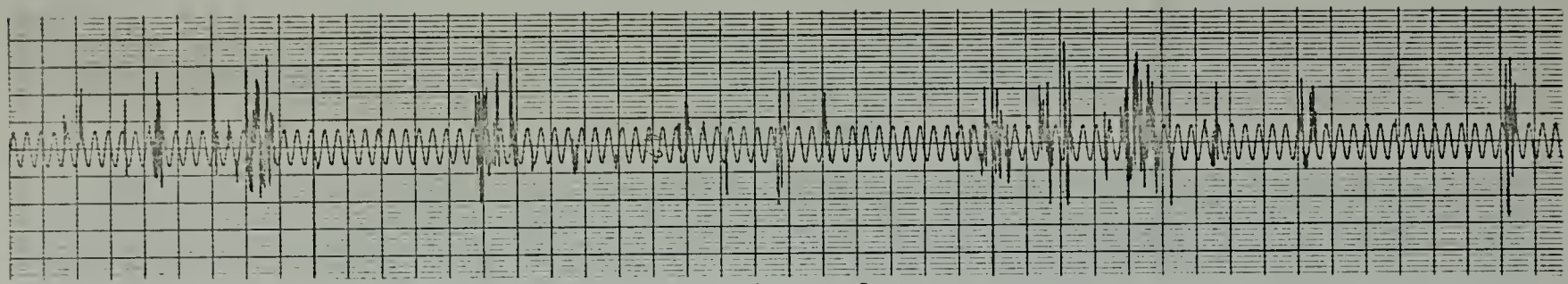
The analysis of the transformations presented in this section lead quite conclusively to the result that the modified FFT is capable of providing better S/N output ratios than the FFT on this type of data. The modified FFT provides substantially improved S/N output ratios over the FFT in Figures 2-18, 2-19, and 2-20. In Figures 2-21 neither transformation provides sufficient signal recognition.

Figures 2-22 through 2-24 are photographs of the spectra presented in Figures 2-18 through 2-20. The photographs emphasize the extra noise smoothing provided by the modified FFT. Figures 2-25 through 2-28 result from the average of 80 transforms instead of 40 as was done in the previous figures. In each of these figures the variance of the noise coefficients is considerably greater in the FFT spectrum than in the modified FFT spectrum. In each of the figures the signal is more distinct and more easily recognized in the modified FFT spectrum than in the FFT spectrum.

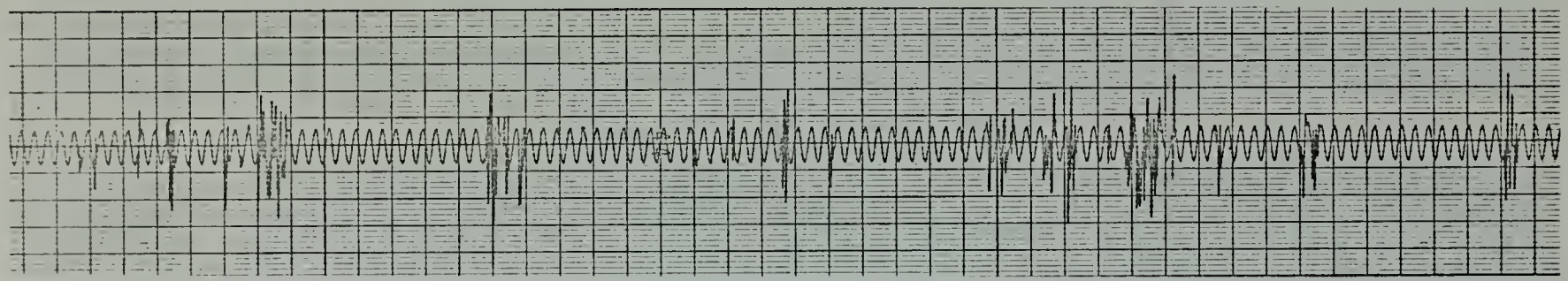
Table IV shows the consistently better S/N output ratios of the modified FFT over the FFT. The -24 db input S/N ratio is the only exception. Another interesting observation

is the improvement of the S/N output ratios obtained by multiplying the FFT and modified FFT spectra together. The product of the two spectra is consistently better than either the individual spectra or the FFT spectrum squared.

Table V further emphasizes the modified FFT's merits in transforming this data. By averaging 80 transforms the modified FFT was able to produce better S/N output ratios than the FFT in every case tested.



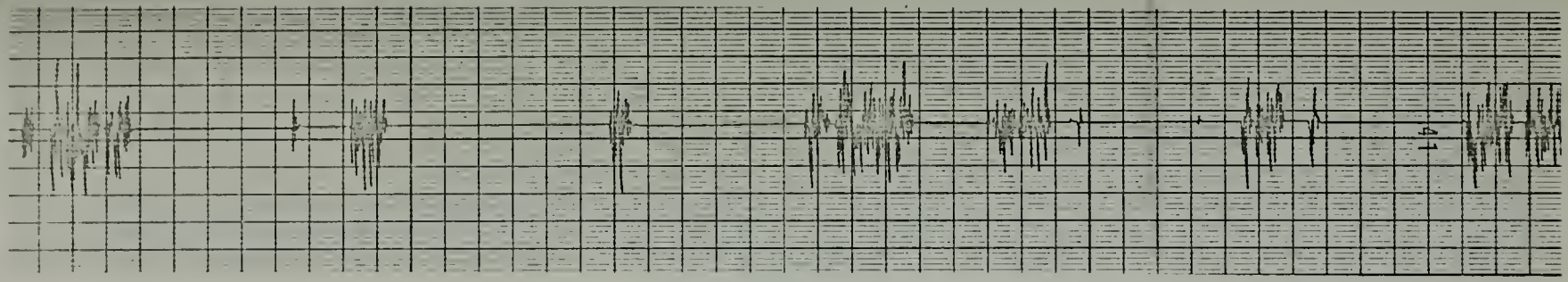
Unfiltered



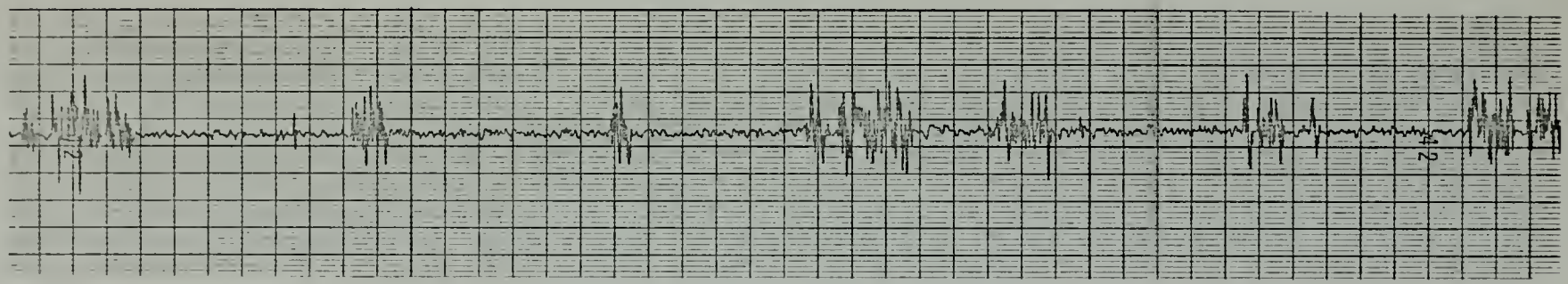
Filtered at 100 hz

10 hz Sinusoid with Burst Noise

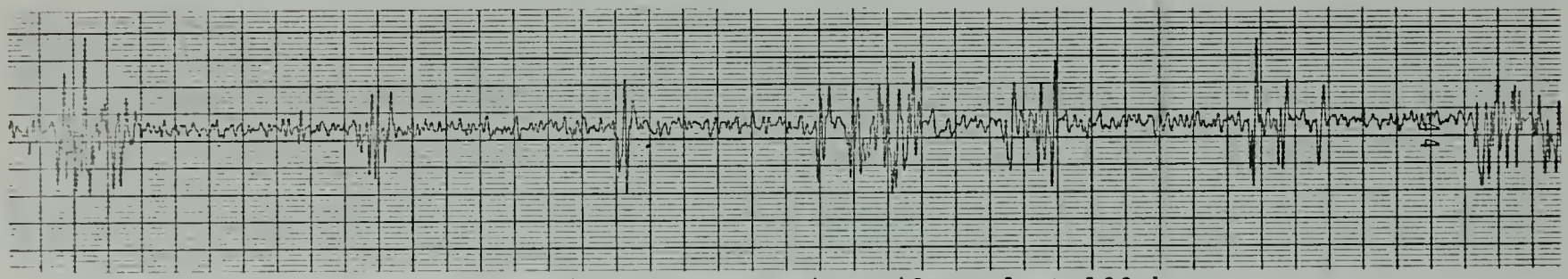
Figure 2-15



Burst Noise



Signal + Noise + Burst Noise

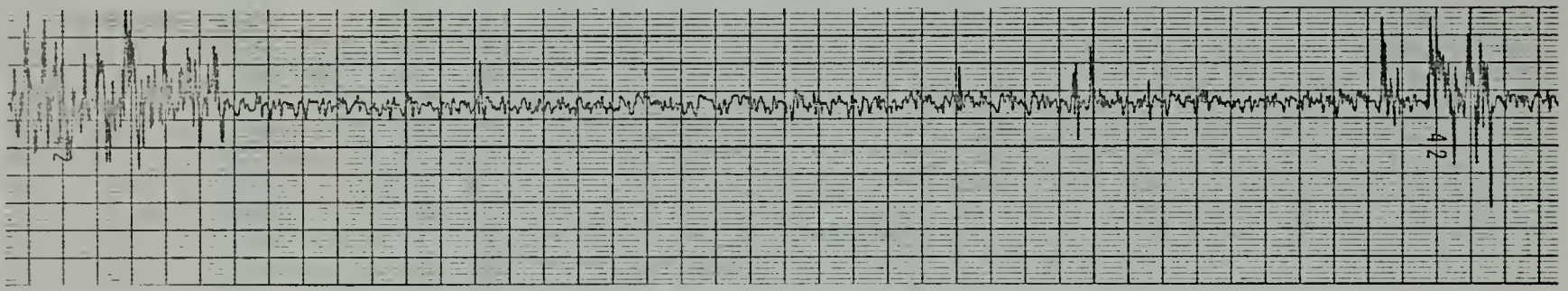


Signal + Noise + Burst Noise Filtered at 100 hz

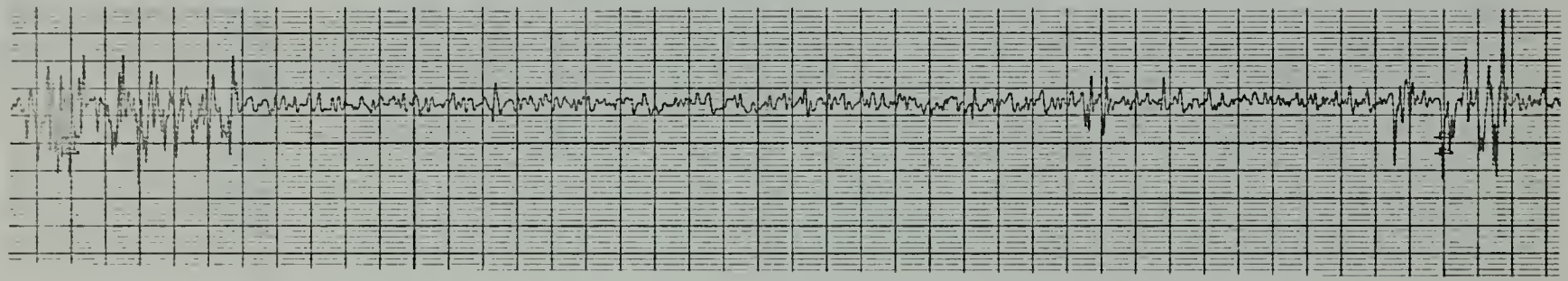
10 hz Sinusoid + Gausian Noise + Burst Noise, S/N = -12 db

Figure 2-16

Burst Noise



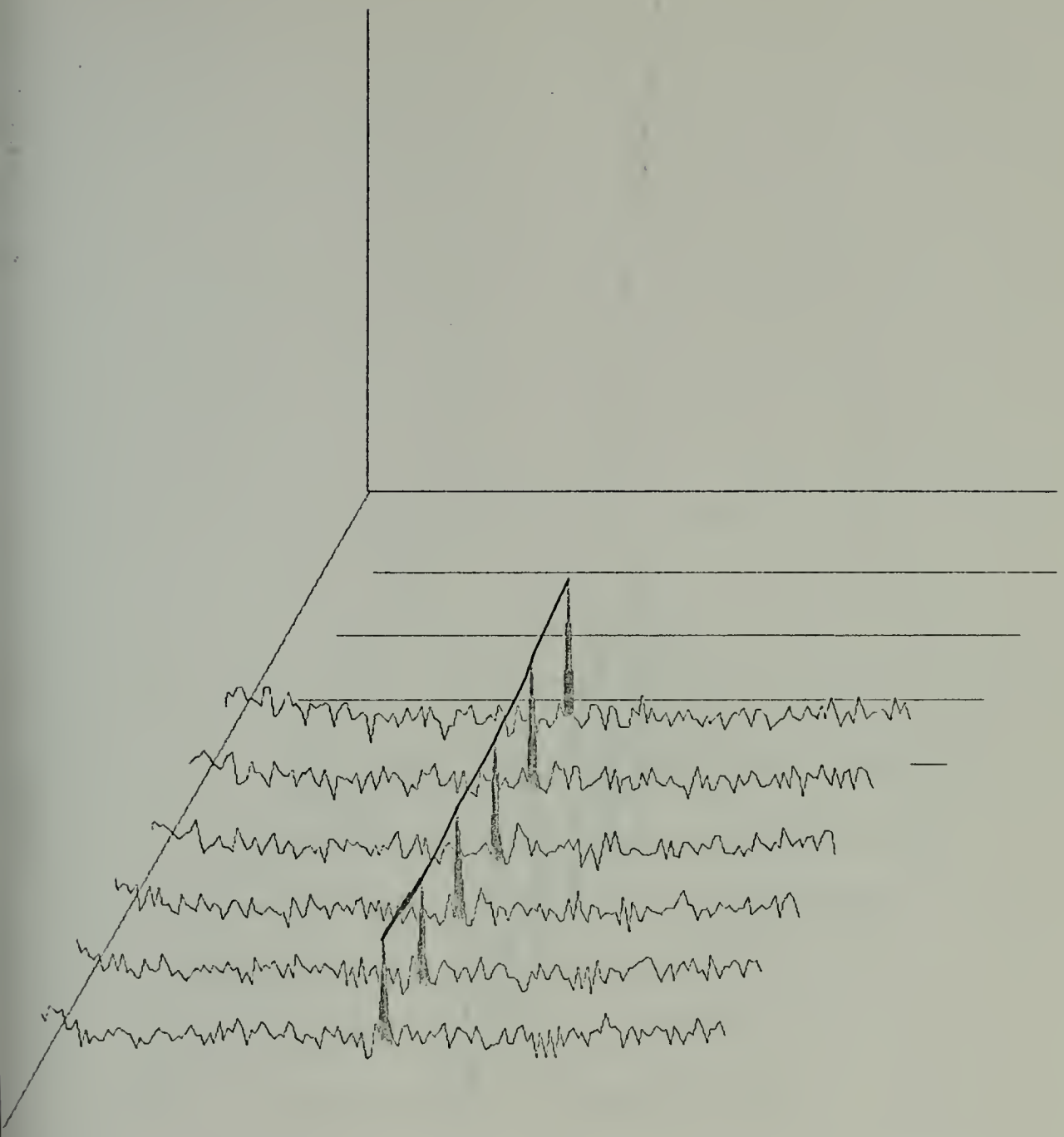
Signal + Noise + Burst Noise



Signal + Noise + Burst Noise Filtered at 100 hz

10 hz Sinusoid + Gausian Noise + Burst Noise, S/N = -18 db

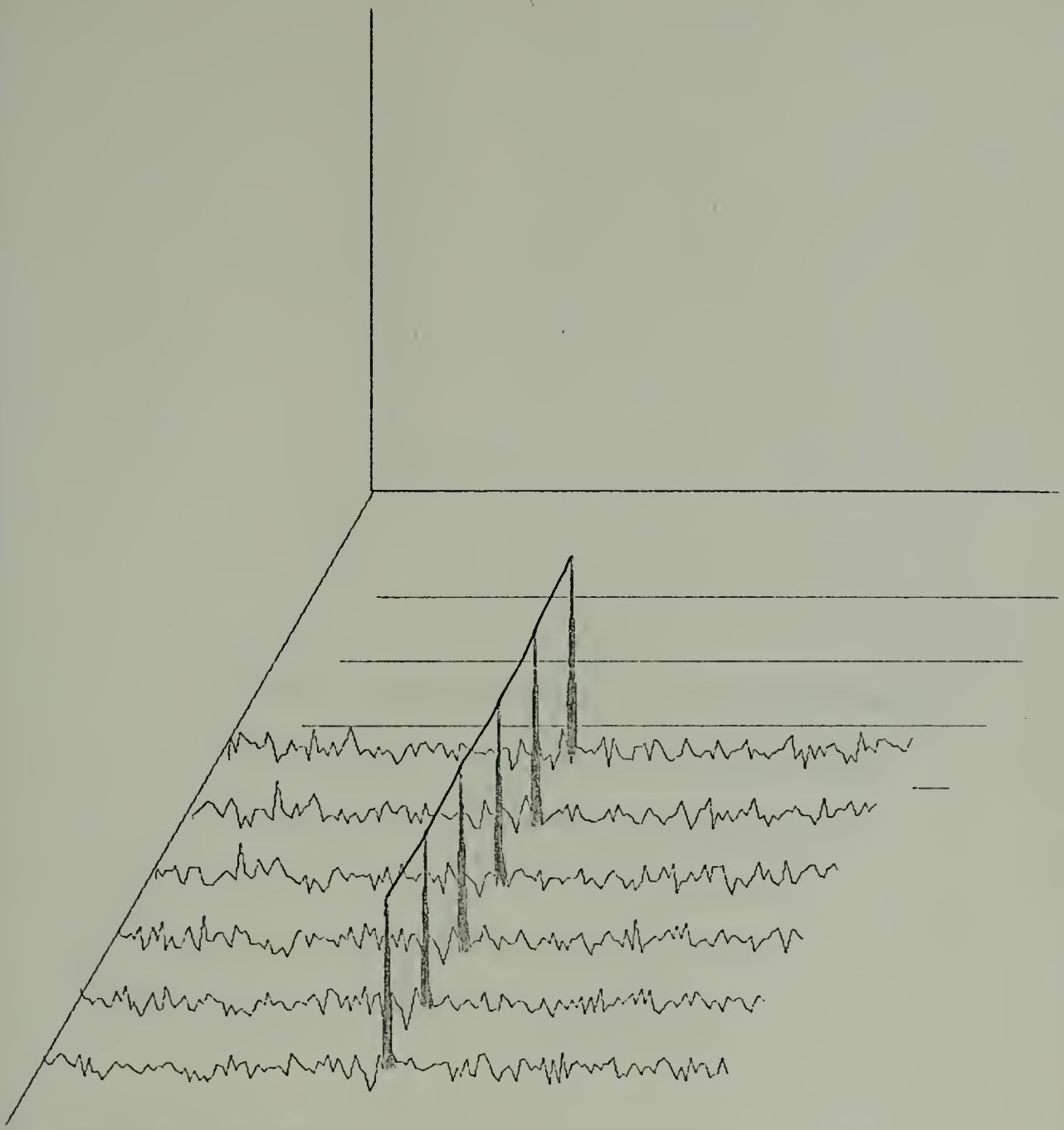
Figure 2-17



40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

X-SCALE = 3.93E-01 UNITS/INCH.
Y-SCALE = 3.80E-01 UNITS/INCH.
N+BURST FFT
N=-12 DB

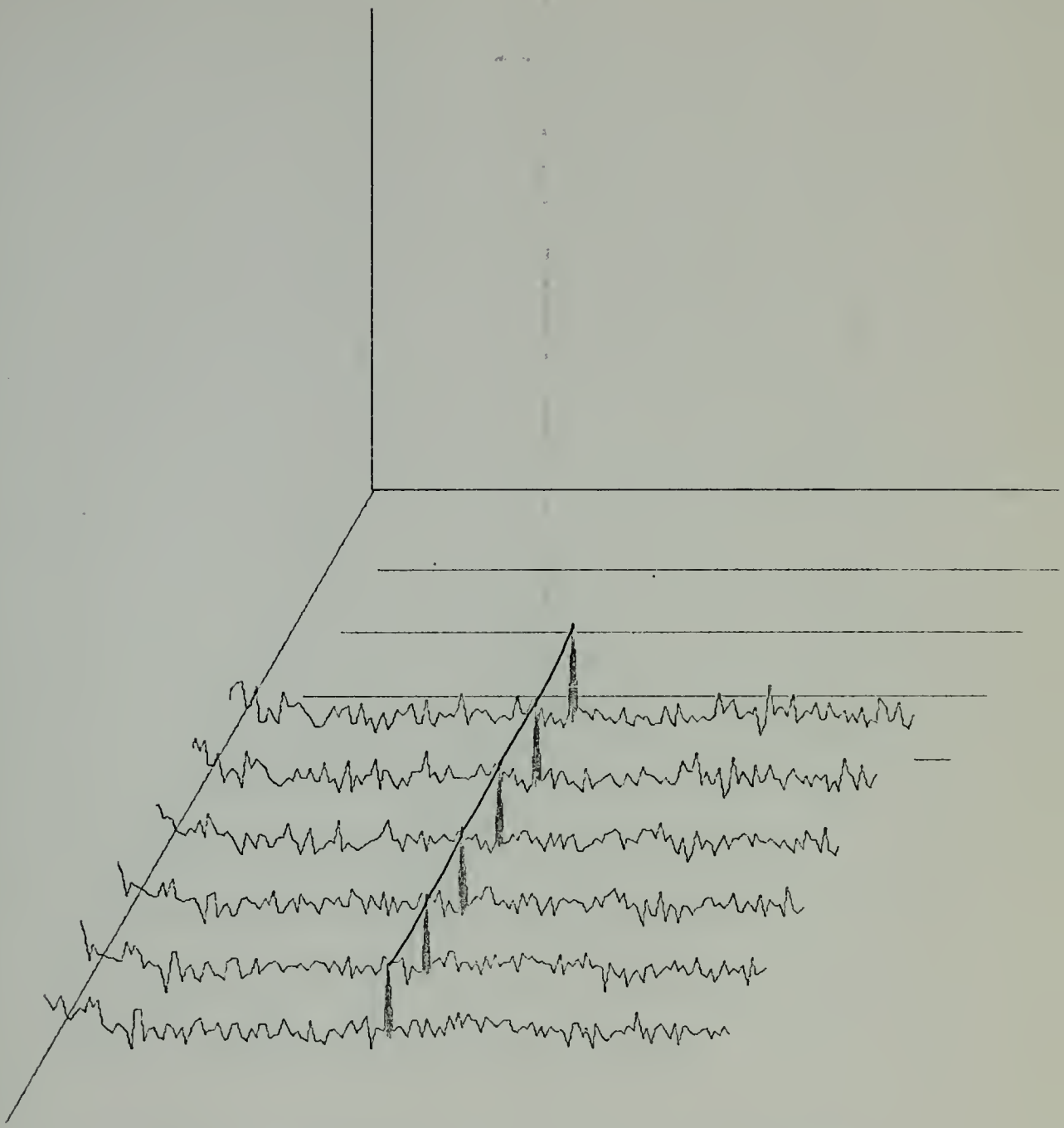
Figure 2-18 (FFT)



40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

X-SCALE = 3.00E-01 UNITS/INCH.
Y-SCALE = 3.00E-01 UNITS/INCH.
S+N+BURST MOD FFT
S/N= -12 DB

Figure 2-18 (Mod-FFT)

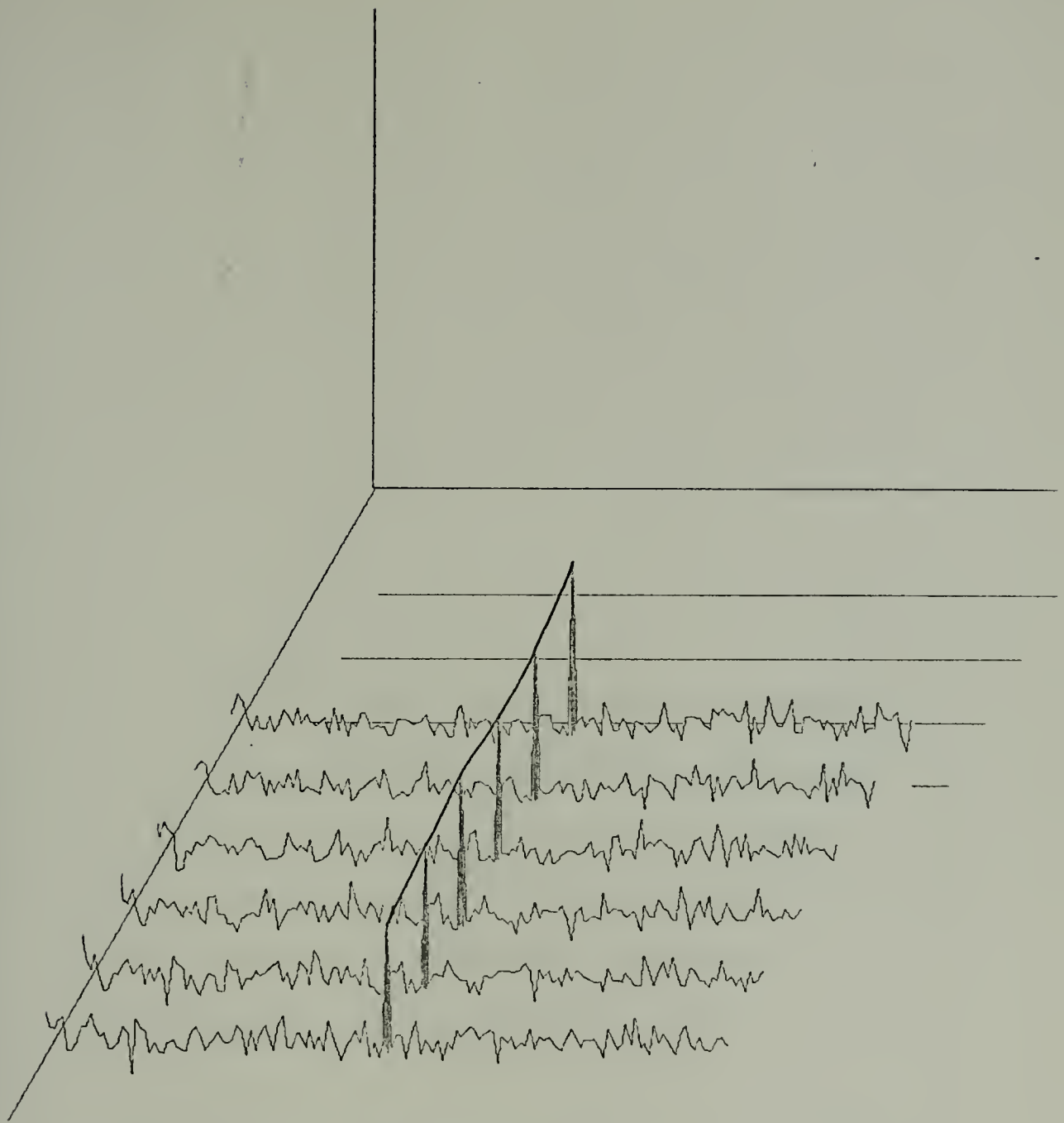


40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

X-SCALE = 3.00E-01 UNITS/INCH.
Y-SCALE = 3.00E-01 UNITS/INCH.

S+N+BURST FFT
S/N= -18 DB

Figure 2-19 (FFT)



40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

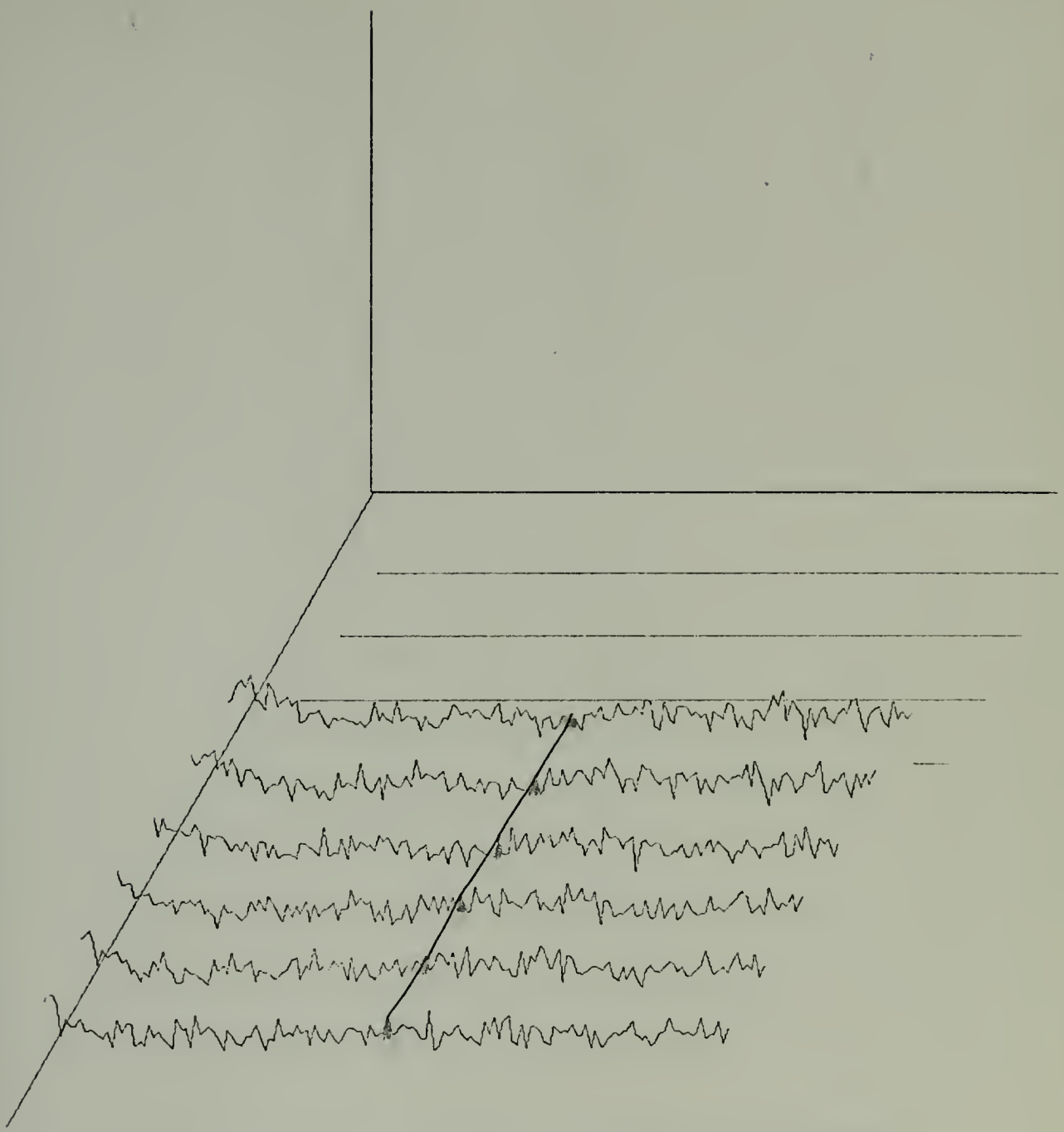
S+N+Burst Mod FFT

X-SCALE = 3.00E-01 UNITS/INCH.

Y-SCALE = 3.00E-01 UNITS/INCH.

S/N= -18 DB

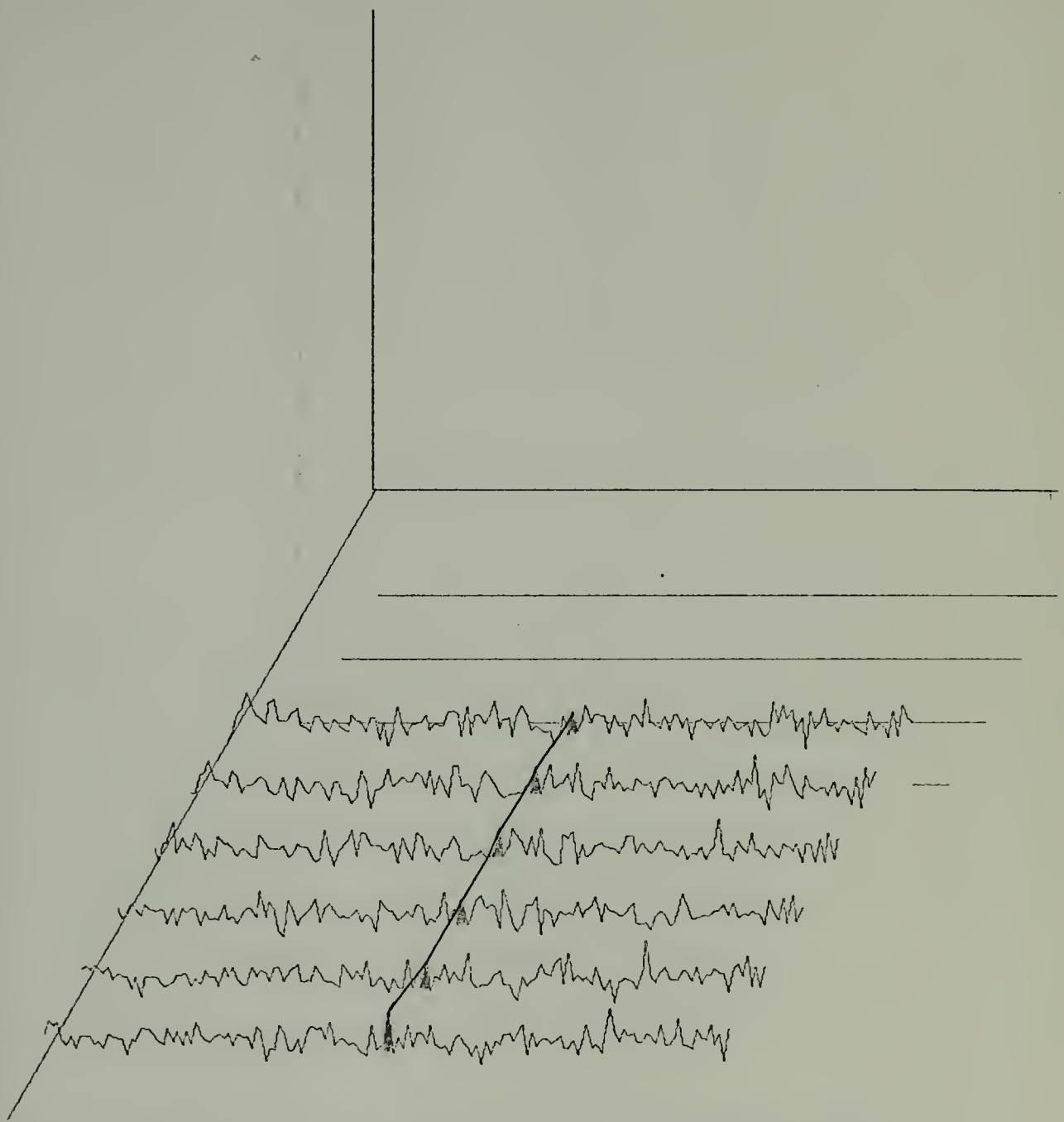
Figure 2-19 (Mod-FFT)



40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

X-SCALE = 3.00E-01 UNITS/INCH.
Y-SCALE = 3.00E-01 UNITS/INCH.
S+N+BURST FFT
S/N= -24 DB

Figure 2-20 (FFT)



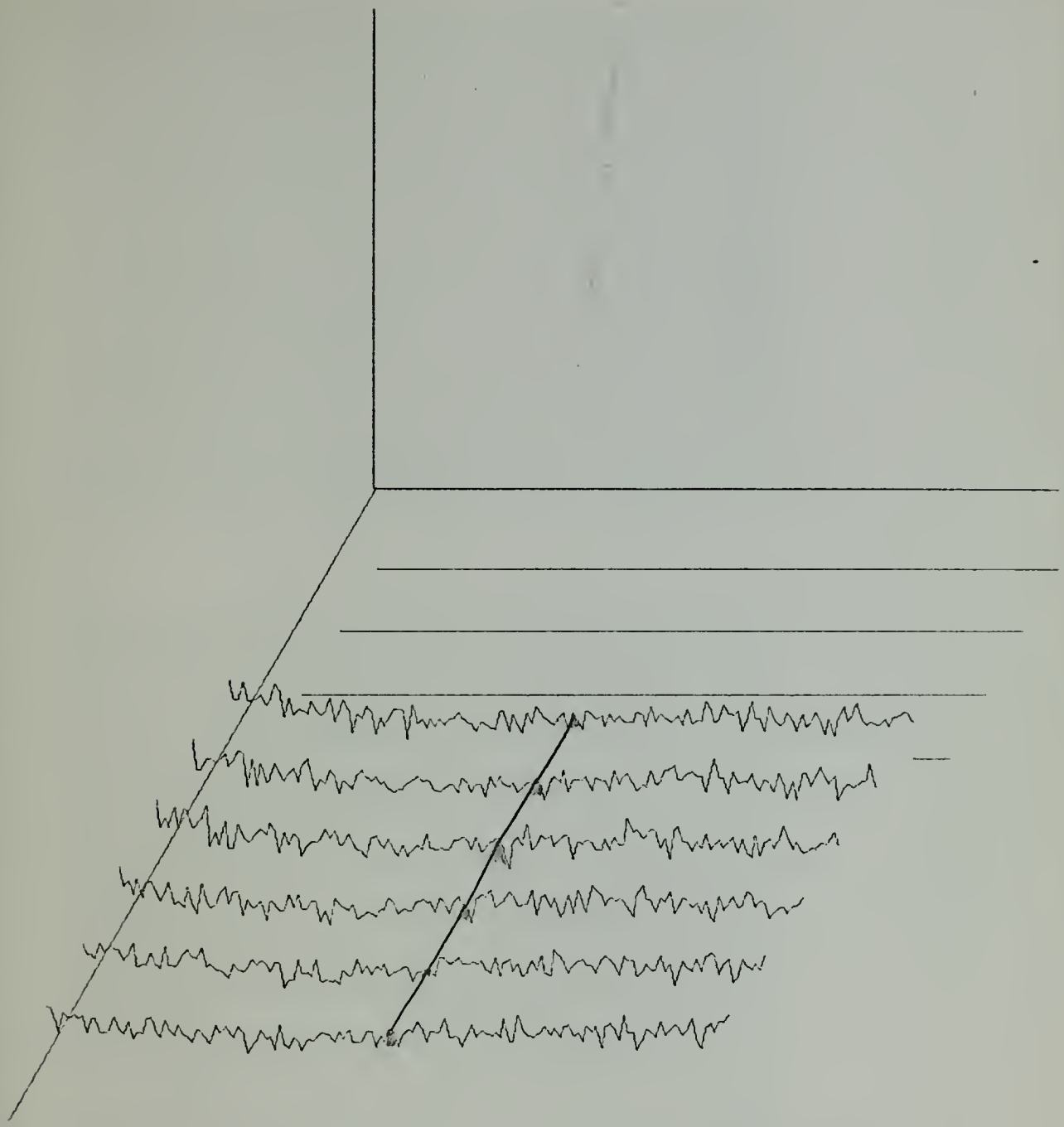
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

S+N+Burst Mod FFT

X-SCALE = 3.00E-01 UNITS/INCH.
Y-SCALE = 3.00E-01 UNITS/INCH.

S/N= -24 DB

Figure 2-20 (Mod-FFT)

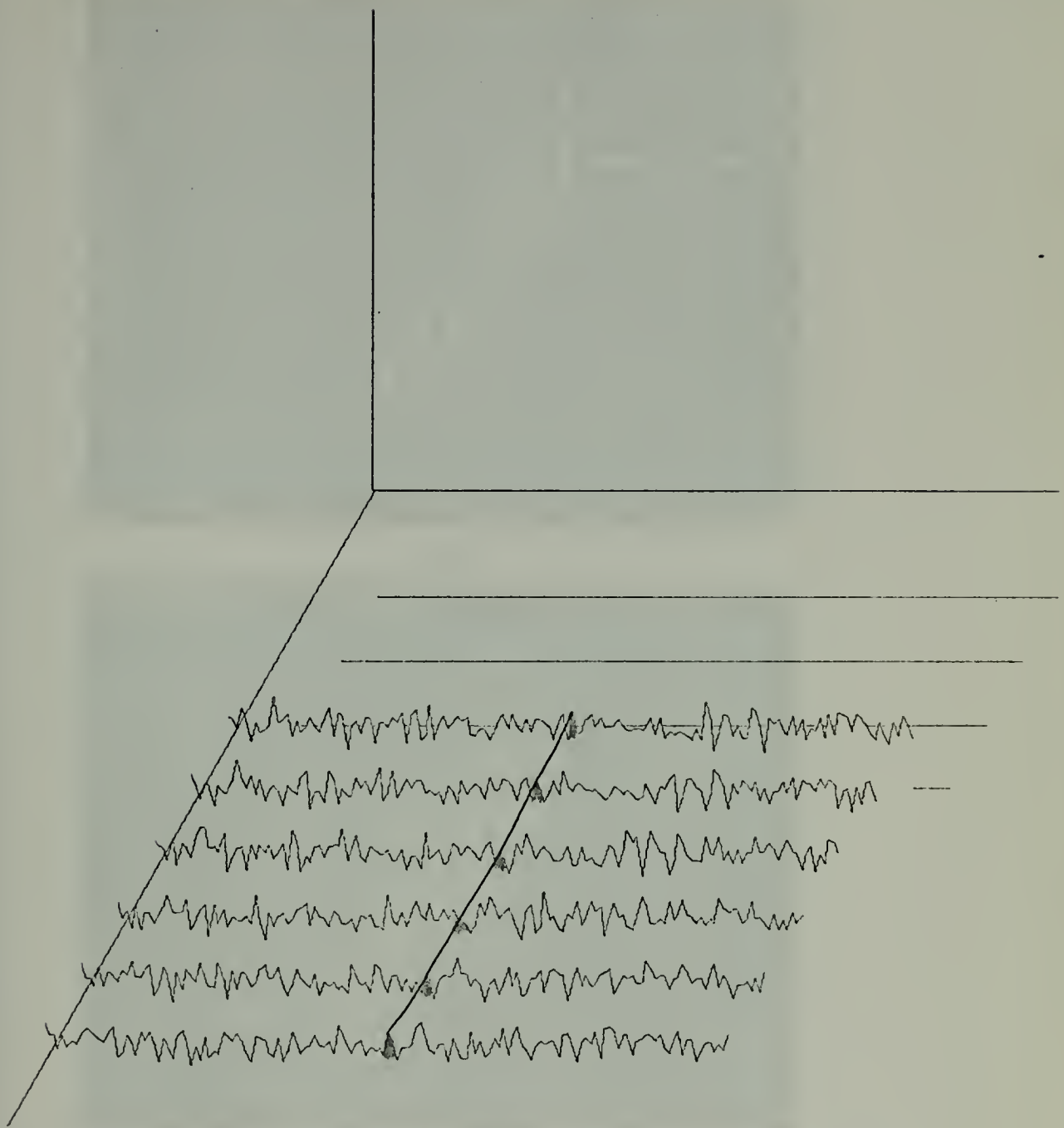


40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

X-SCALE = 3.88E-01 UNITS/INCH.
Y-SCALE = 3.88E-01 UNITS/INCH.

S+N+BURST FFT
S/N=-30 DB

Figure 2-21 (FFT)



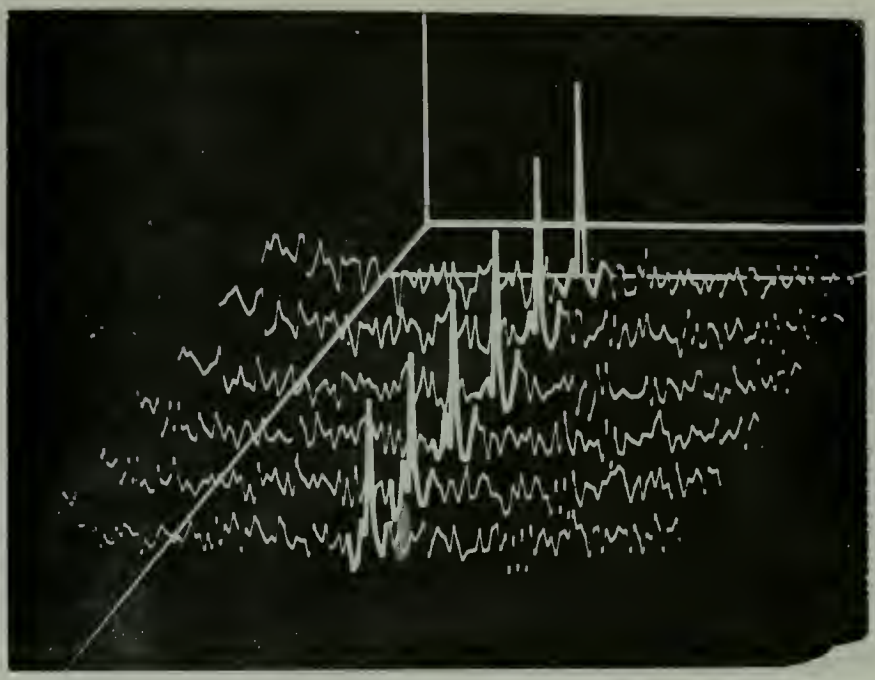
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

S+N+Burst Mod FFT

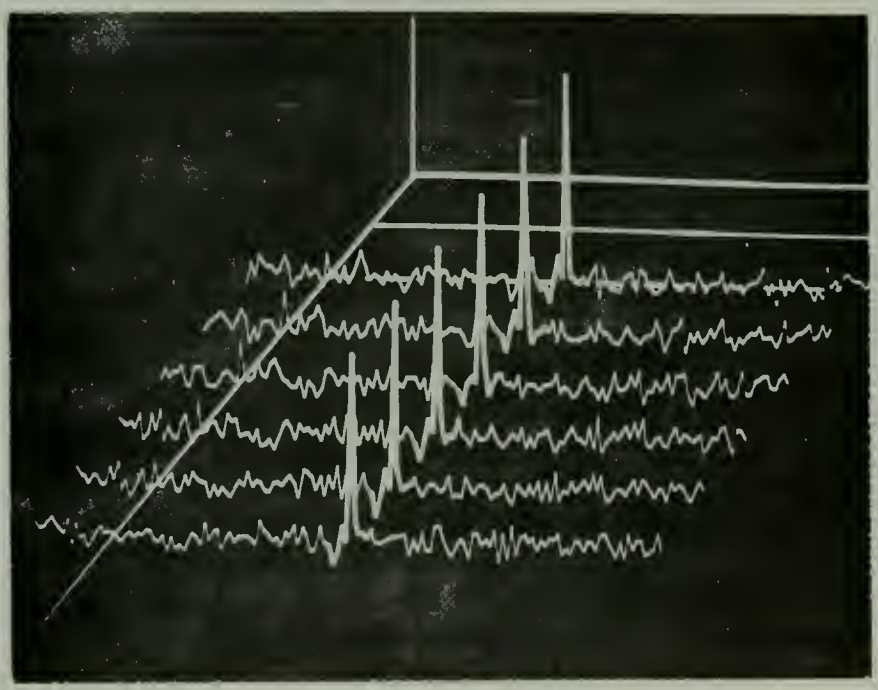
X-SCALE = 3.00E-01 UNITS/INCH.
Y-SCALE = 3.00E-01 UNITS/INCH.

S/N=-30 DB

Figure 2-21 (Mod-FFT)



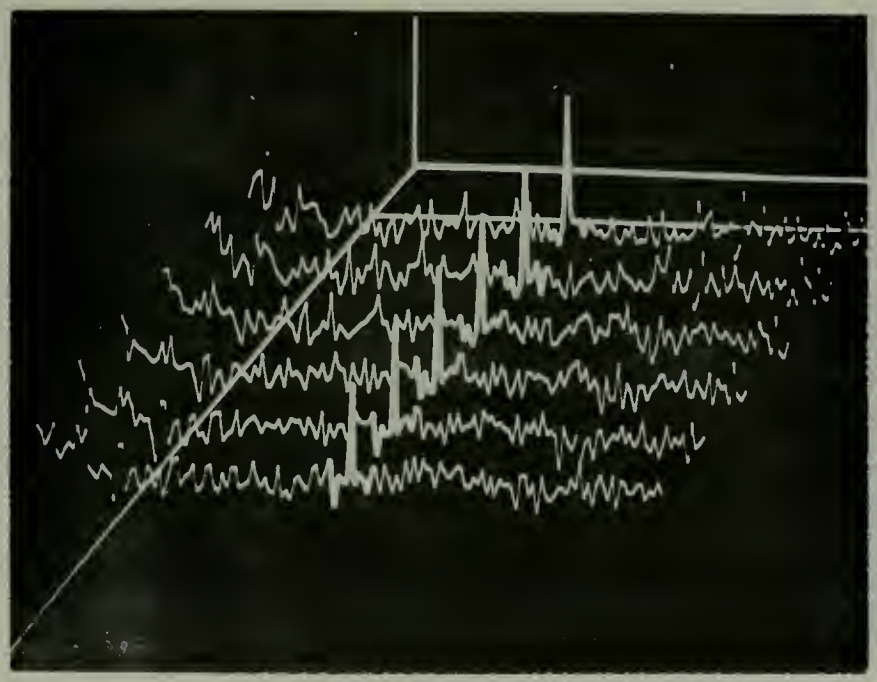
FFT



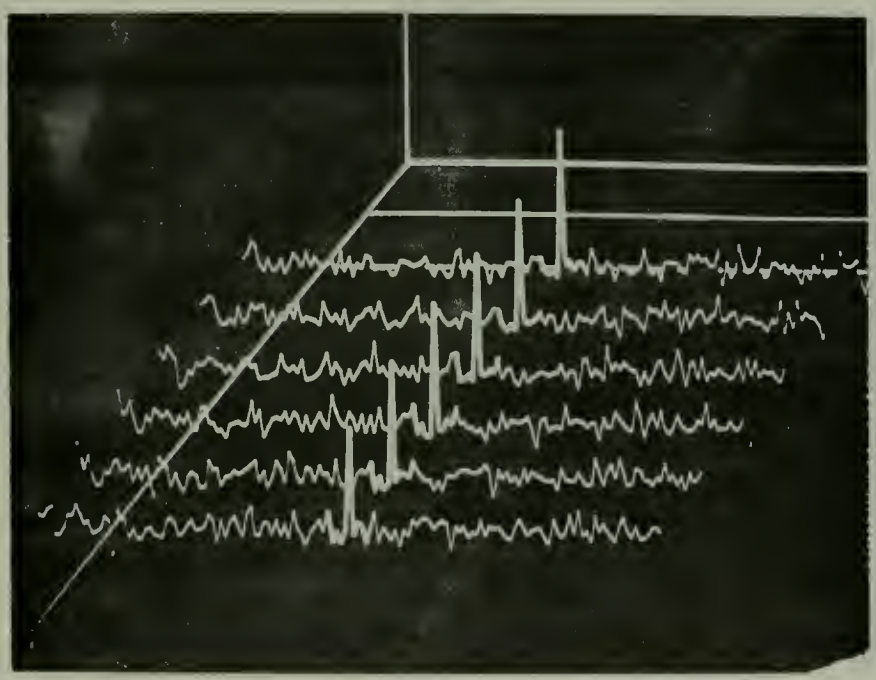
Modified FFT

10 hz signal, S/N = -12 db + burst noise
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

Figure 2-22



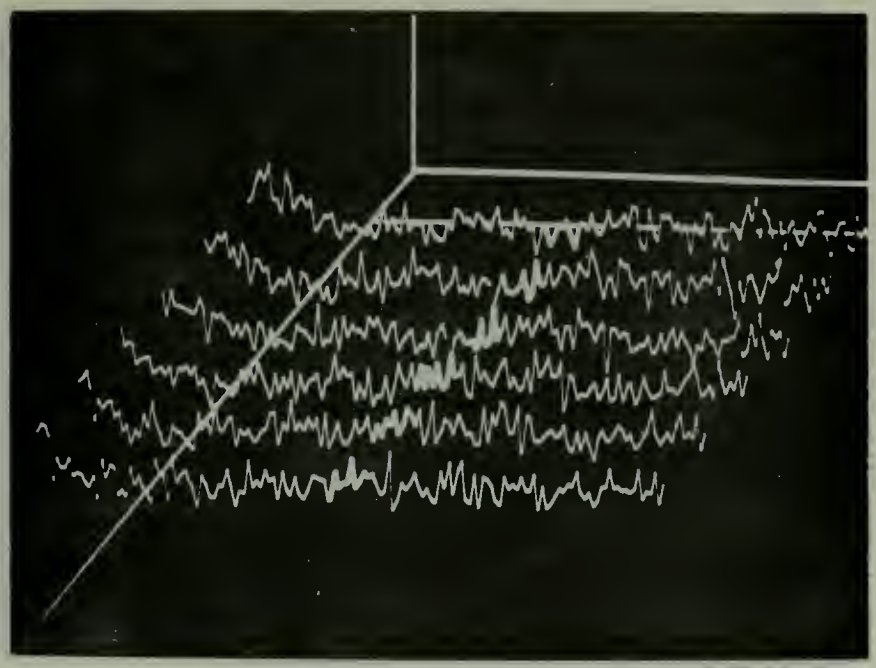
FFT



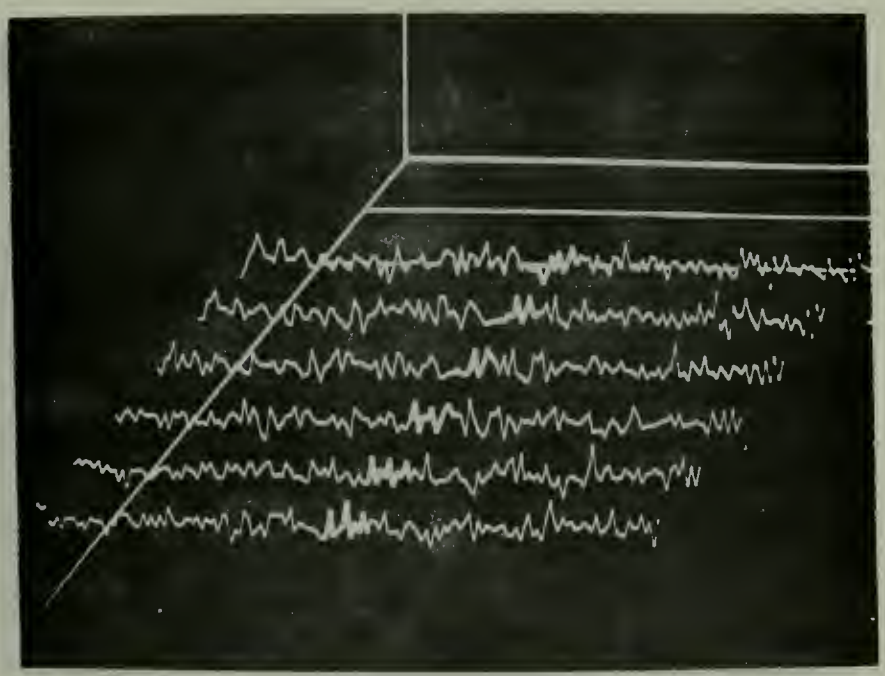
Modified FFT

10 hz signal, S/N = -18 db + burst noise
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

Figure 2-23



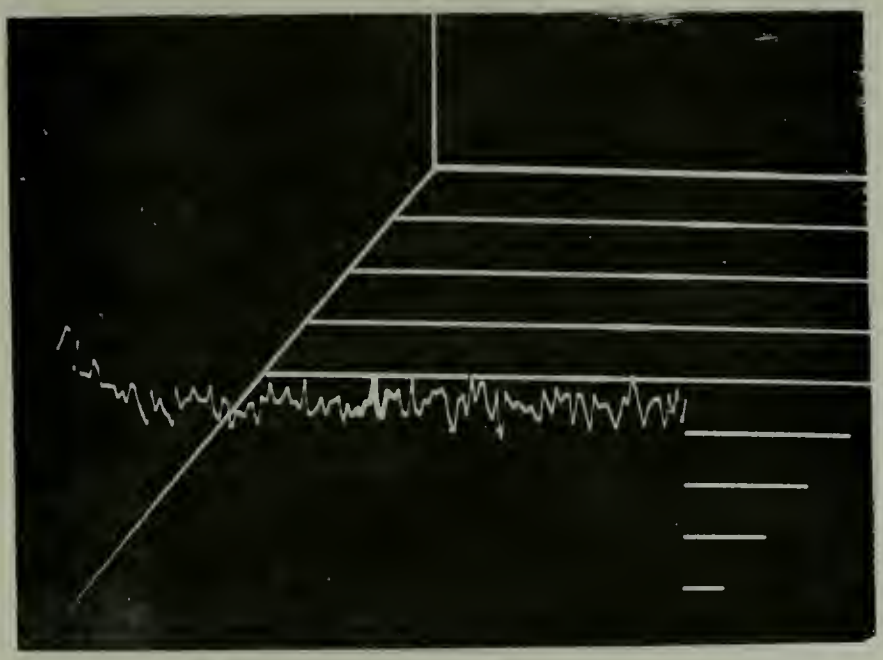
FFT



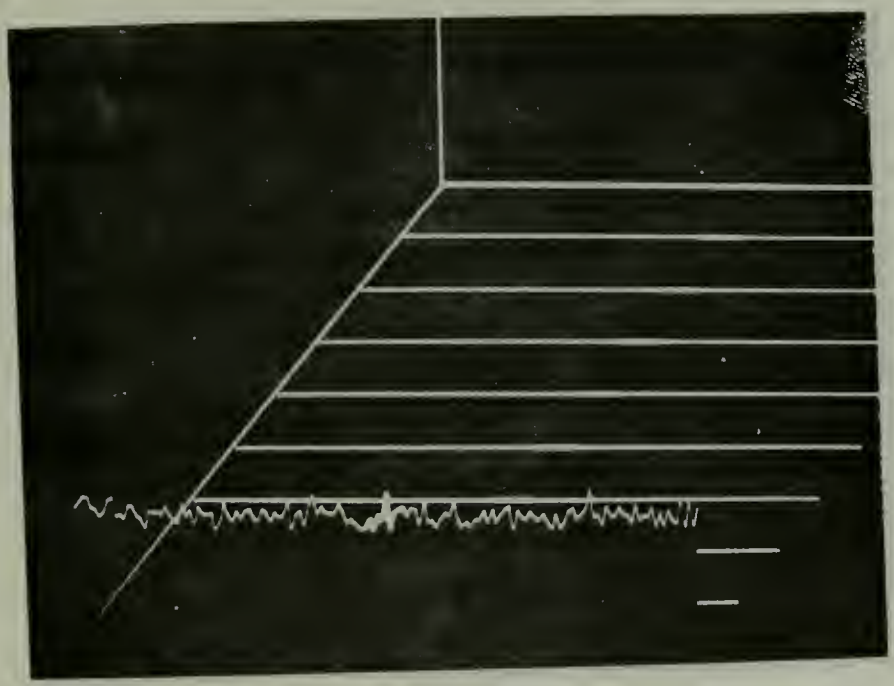
Modified FFT

10 hz signal, S/N = -24 db + burst noise
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

Figure 2-24



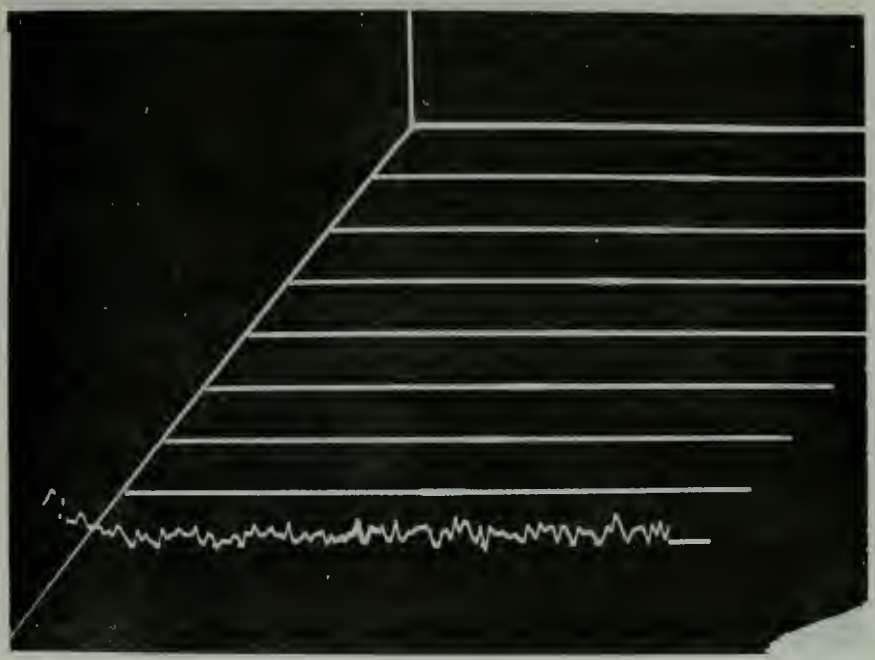
FFT



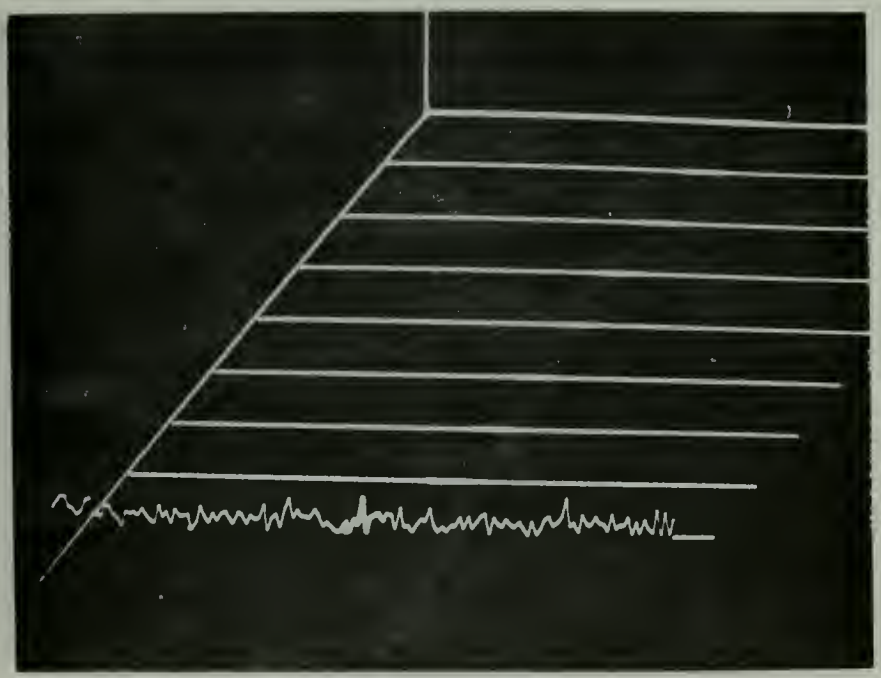
Modified FFT

10 hz signal, S/N = -24 db + burst noise
80 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

Figure 2-25



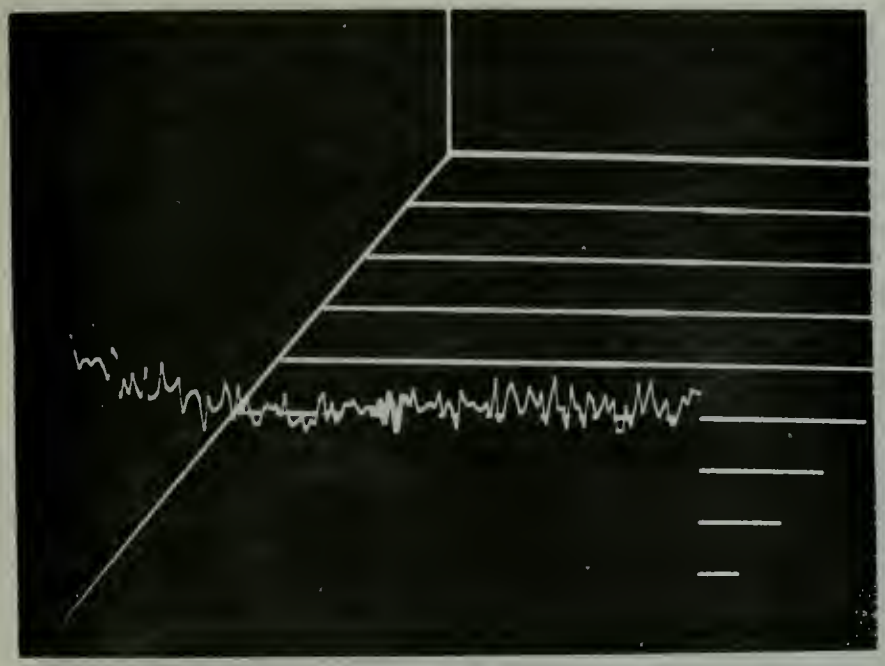
FFT



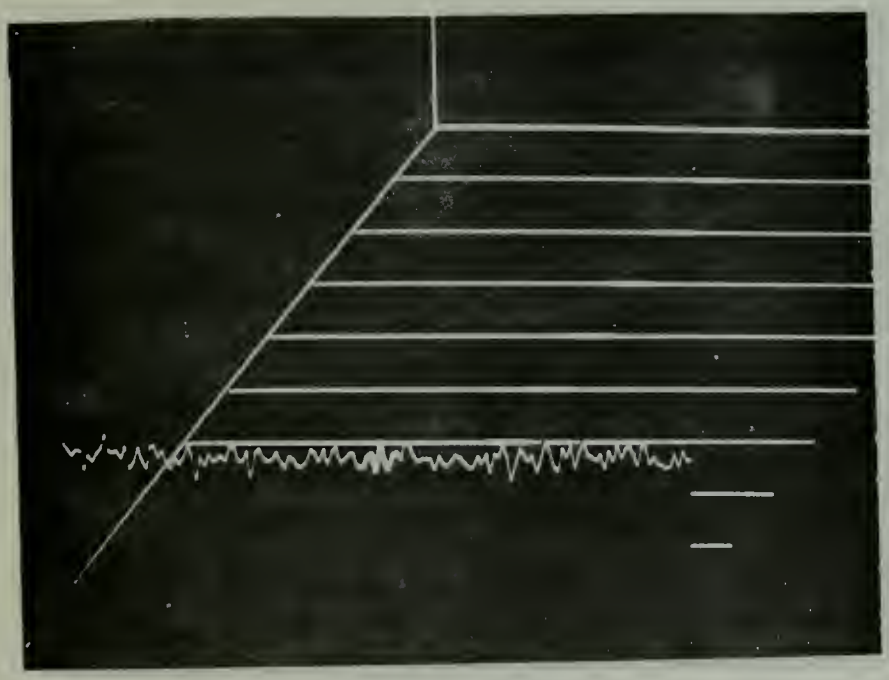
Modified FFT

10 hz signal, S/N = -24 db + burst noise
80 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution
Spectrum squared

Figure 2-26



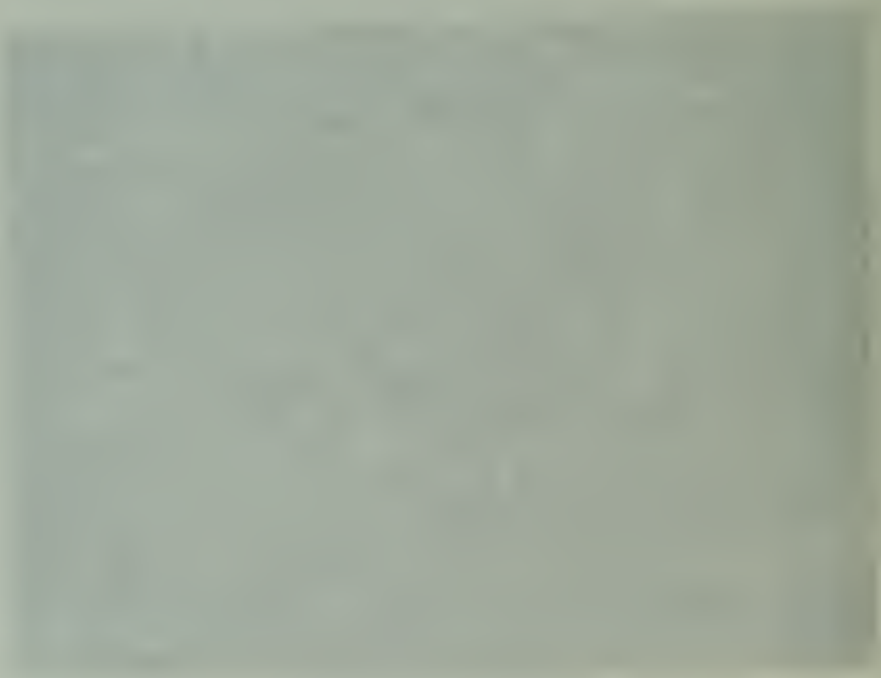
FFT

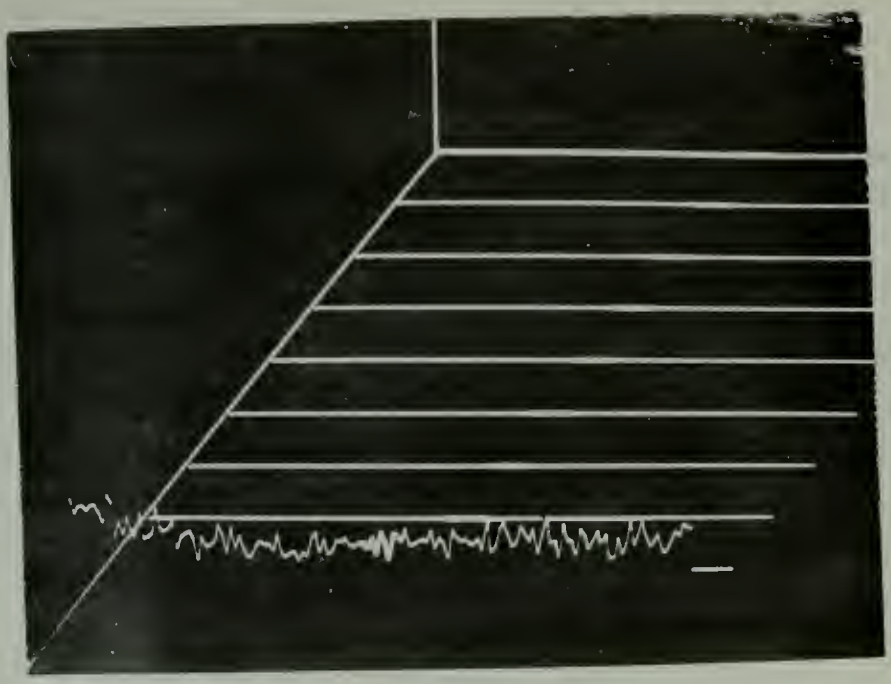


Modified FFT

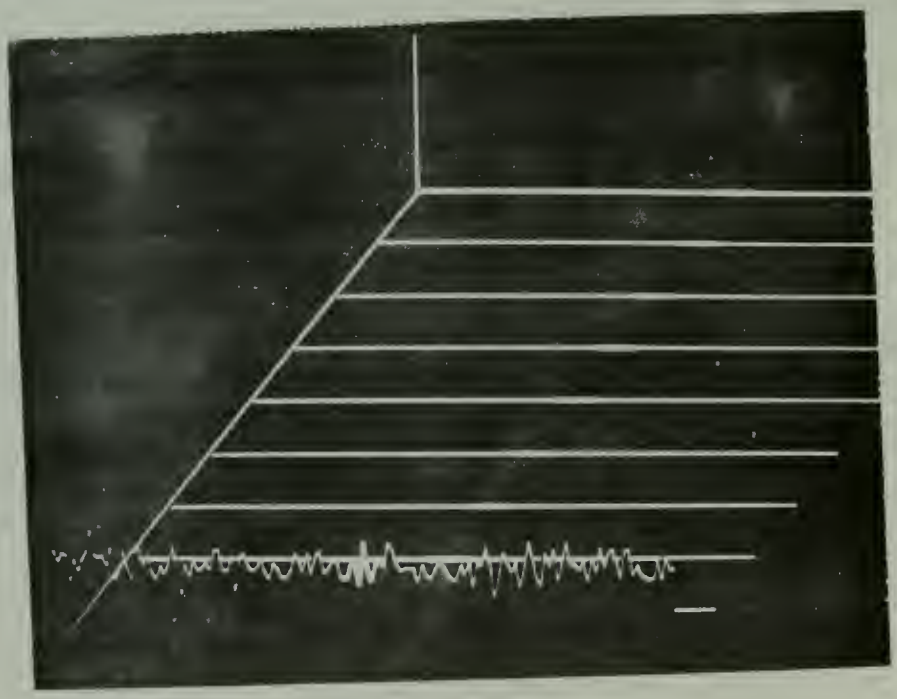
10 hz signal, S/N = -30 db + burst noise
80 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution

Figure 2-27





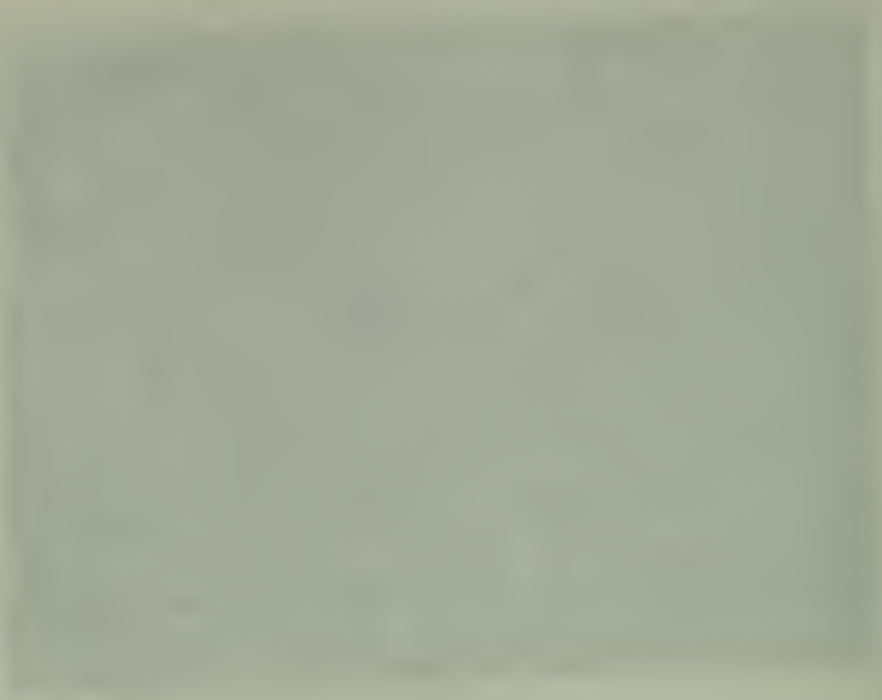
FFT



Modified FFT

10 hz signal, S/N = -30 db + burst noise
80 transforms averaged per trace, transform length 2048
Displayed bandwidth 18.75 hz, 0.125 hz resolution
Spectrum squared

Figure 2-28



	S/N out (db)			
S/N in (db)	FFT	Mod-FFT	FFT×FFT	FFT× Mod-FFT
-12	13.770	22.354	16.342	23.711
-18	7.729	14.389	8.671	15.133
-24	8.747	8.360	6.450	8.863
-30	5.109	5.515	3.551	5.910

Table IV

Average S/N out (db) vs. S/N in (db) for the Average of 40 Transforms for the case of the 10 hz signal plus Gaussian noise plus burst noise. S/N out computed as:

$$\frac{(F_s-\mu_n)^2}{2\sigma_n^2}$$

where F_s is the magnitude of the signal coefficient and μ_n and σ_n are computed from 25 noise coefficients on either side of F_s.

	S/N out (db)			
S/N in (db)	FFT	Mod-FFT	FFT×FFT	FFT× Mod-FFT
-12	16.154	26.606	18.731	26.945
-18	11.433	16.341	12.855	17.121
-24	9.332	10.979	6.713	10.061
-30	6.875	8.094	5.130	7.961

Table V

Average S/N out (db) vs. S/N in (db) for the Average of 80 Transforms for the case of the 10 hz signal plus Gaussian noise plus burst noise. S/N out computed as:

$$\frac{(F_s-\mu_n)^2}{2\sigma_n^2}$$

where F_s is the magnitude of the signal coefficient and μ_n and σ_n are computed from 25 noise coefficients on either side of F_s.

E. UNDERWATER ACOUSTIC DATA

The final data to be analysed was underwater acoustic data. This data provides a great variety of signals from different sources with different strengths all transmitted simultaneously. It is a challenging test of the signal processor's ability to resolve the many different signals and to perceive weak signals in the presence of strong ones. Only selected portions of the data have been presented in this section.

The data was filtered at 200 hz and sampled at 512 samples per second. Again, 2048 point transforms were employed resulting in a frequency resolution of $\frac{1}{4}$ hz.

Each trace in the pictures is the result of averaging 40 transforms. The FFT spectra are in the top photographs in each figure and the modified FFT spectra are in the lower photographs. The time frame and frequency bandwidth are the same for each photograph in a given figure. The total displayed frequency bandwidth in each trace is 37.5 hz.

Figures 2-29 and 2-30 each depict the same frequency bandwidth but at different times. The main distinction is the modified FFT's ability to detect the highlighted signal in Figure 2-29 before the FFT and to make it clearly recognizable. The large signals to the left and right of the highlighted signal are enhanced better by the FFT. There are also significant differences in the general appearance of the two spectra. Figure 2-30 is presented to show that at a later time the FFT presented the same signal as well as

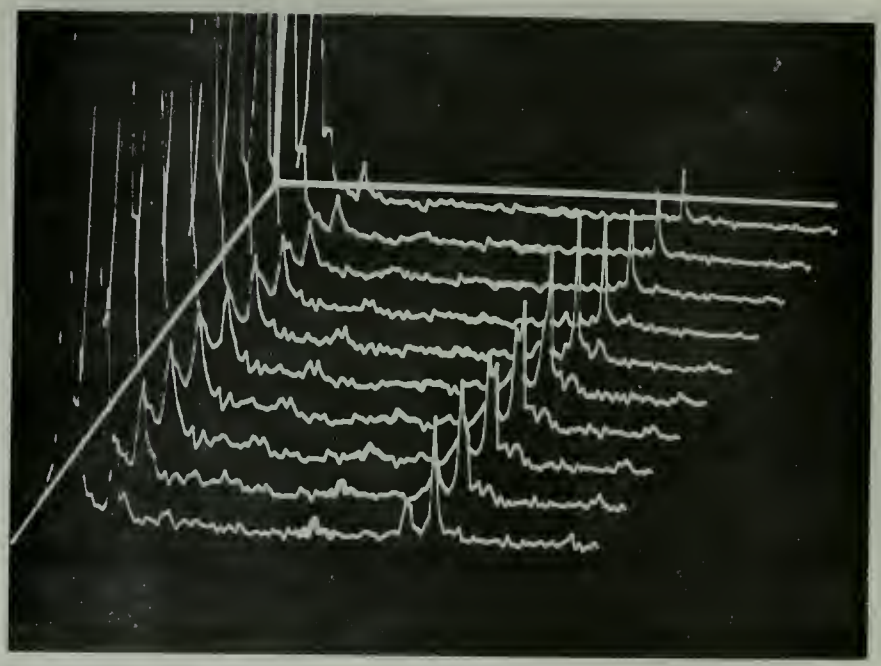
or better than the modified FFT. The signal to the left of the highlighted signal seems to be presented equally well by both spectra.

Figures 2-31 and 2-32 show a relatively weak signal. Close comparison of the traces in the spectra indicate that in some cases the FFT outperforms the modified FFT and in others the reverse is true. The bottom trace of Figure 2-32 provides a dramatic case in which the modified FFT produces the signal better than the FFT.

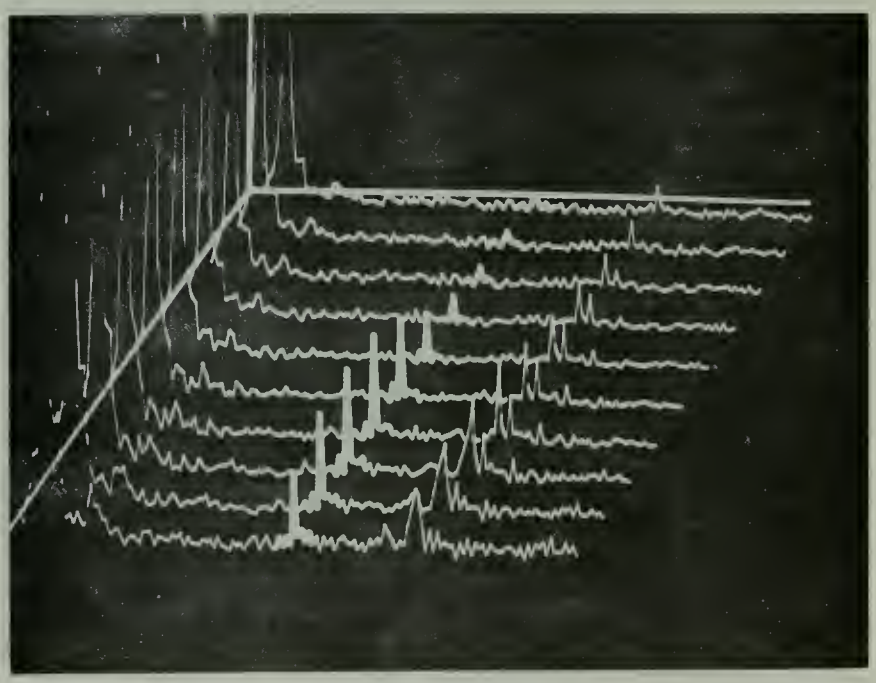
Figures 2-33 through 2-35 provide an example in which the two spectra behave almost identically except for the last trace in Figure 2-35 in which an effect like the one described for Figure 2-32 above occurs.

Figure 2-36 again depicts a signal that has been enhanced better by the modified FFT than by the FFT. Also two distinct signals are present to the right of the highlighted signal in the modified FFT whereas only one is present in the FFT.

Figures 2-37 through 2-39 all show the same signal at different times. In this case the two spectra are again quite similar. The main reason for presenting these spectra is the modified FFT's ability to retain the signal in Figure 2-39 after it has been lost by the FFT.



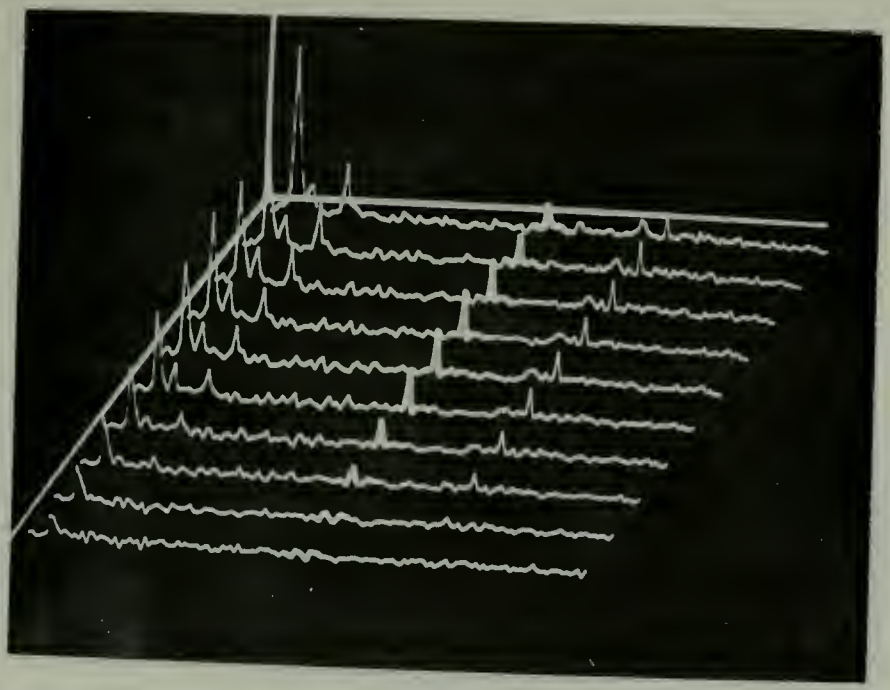
FFT



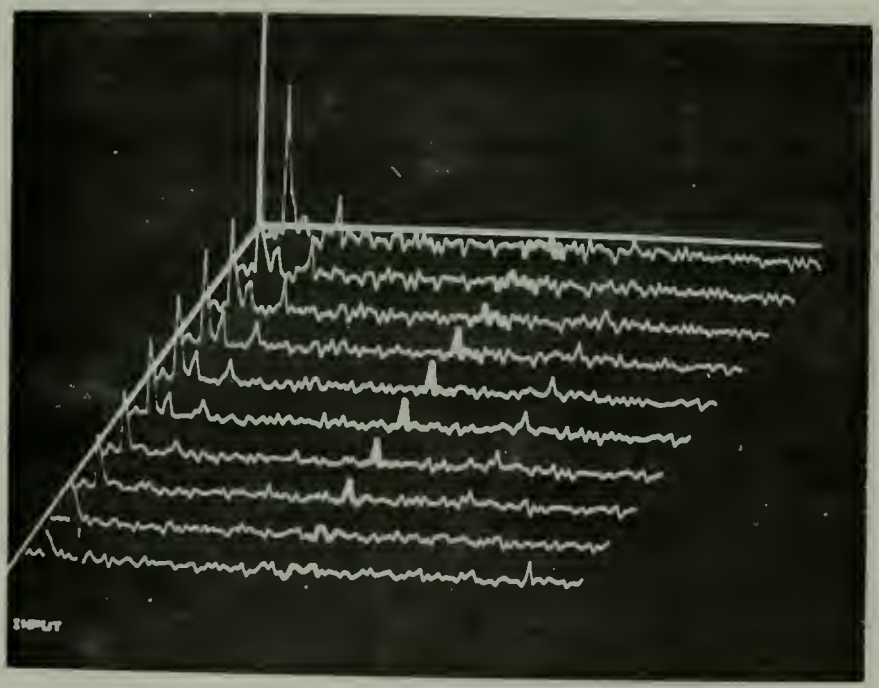
Modified FFT

Underwater Accoustic Data
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 37.5 hz, 0.25 hz resolution

Figure 2-29



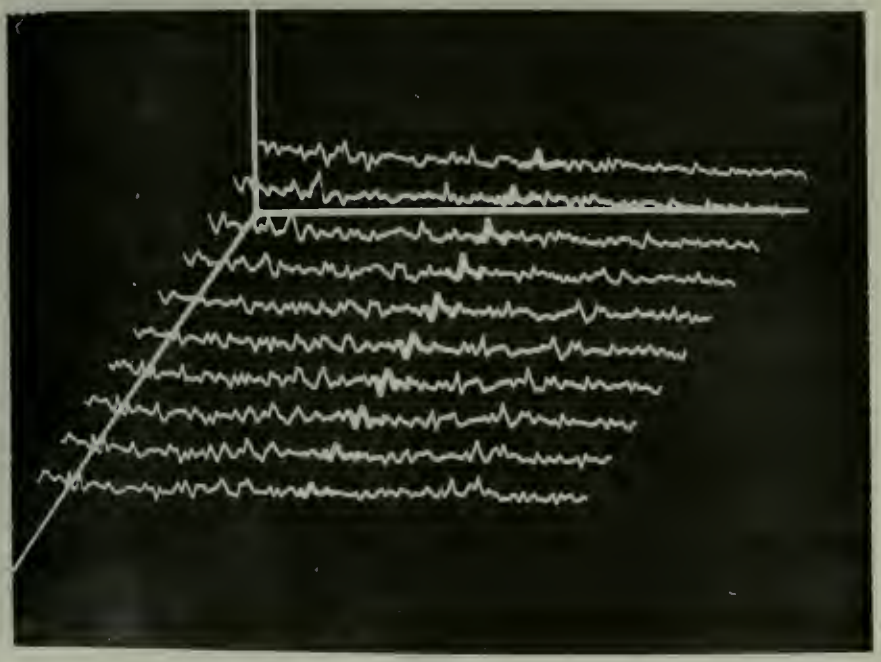
FFT



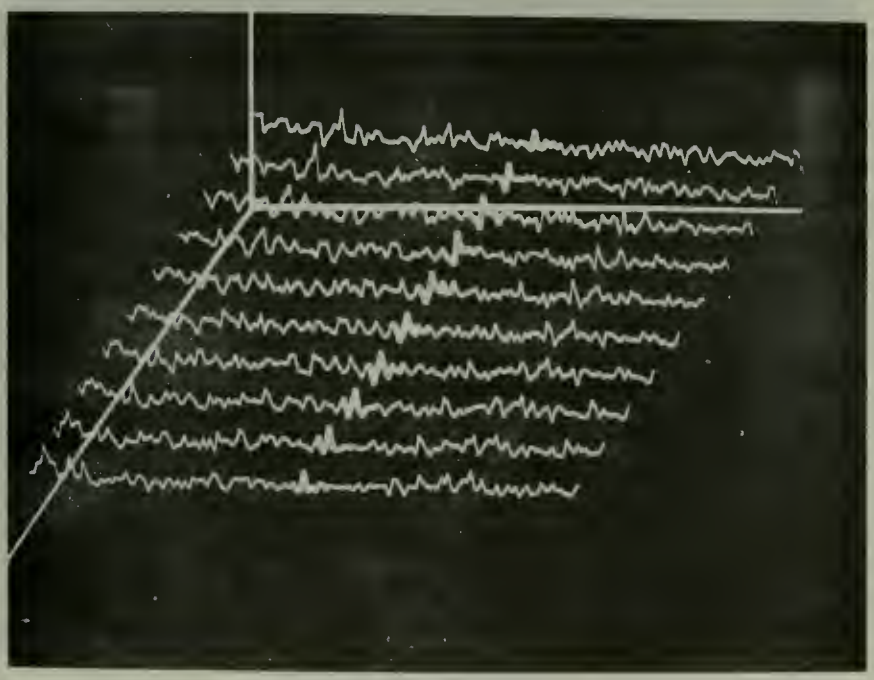
Modified FFT

Underwater Accoustic Data
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 37.5 hz, 0.25 hz resolution

Figure 2-30



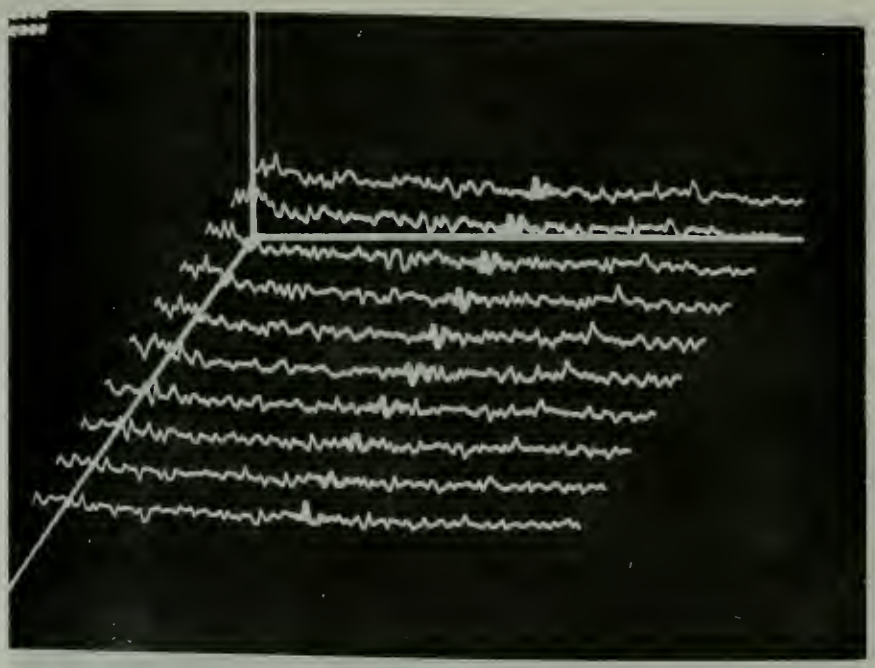
FFT



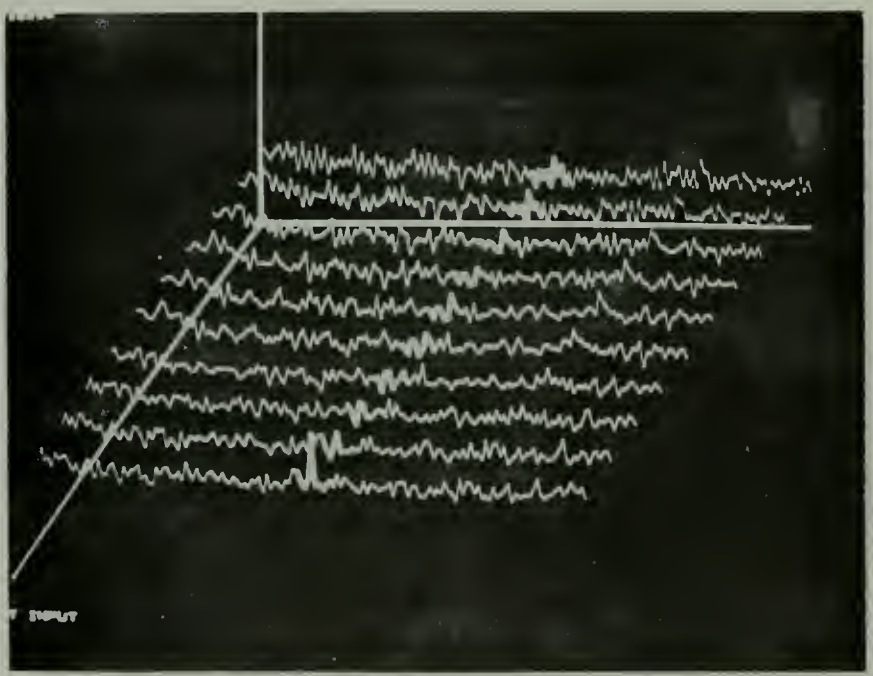
Modified FFT

Underwater Accoustic Data
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 37.5 hz, 0.25 hz resolution

Figure 2-31



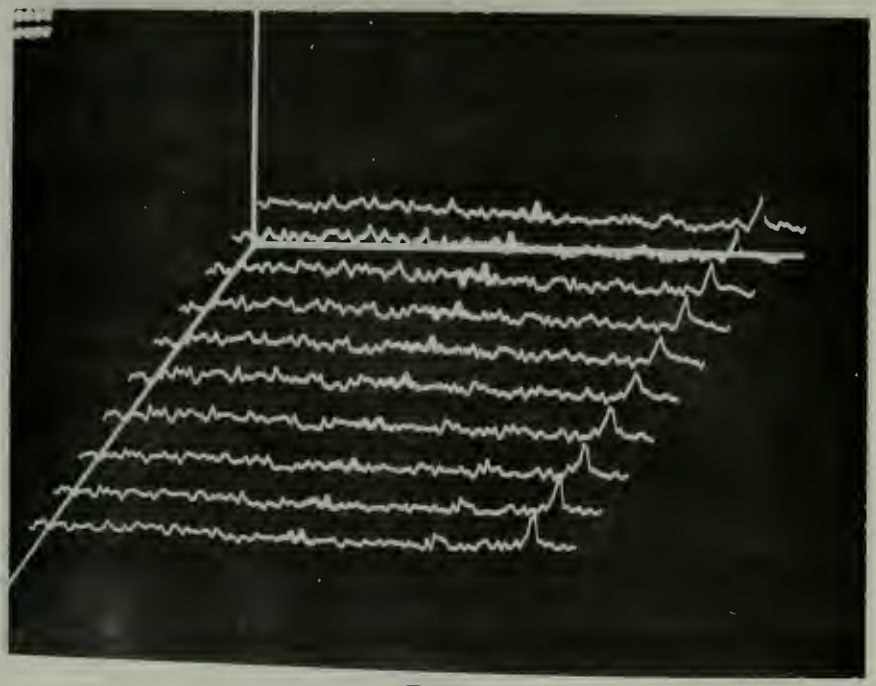
FFT



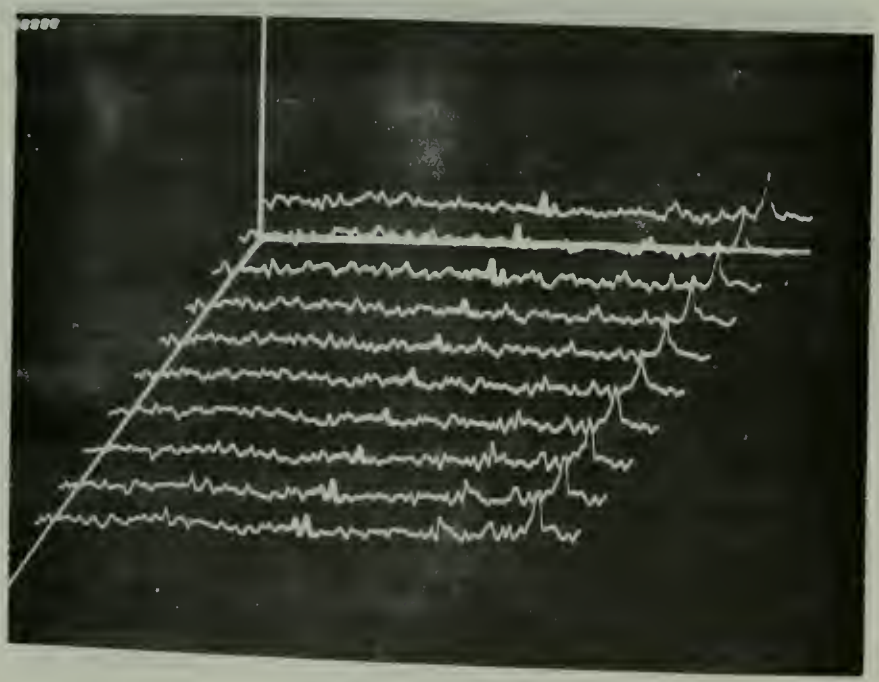
Modified FFT

Underwater Accoustic Data
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 37.5 hz, 0.25 hz resolution

Figure 2-32



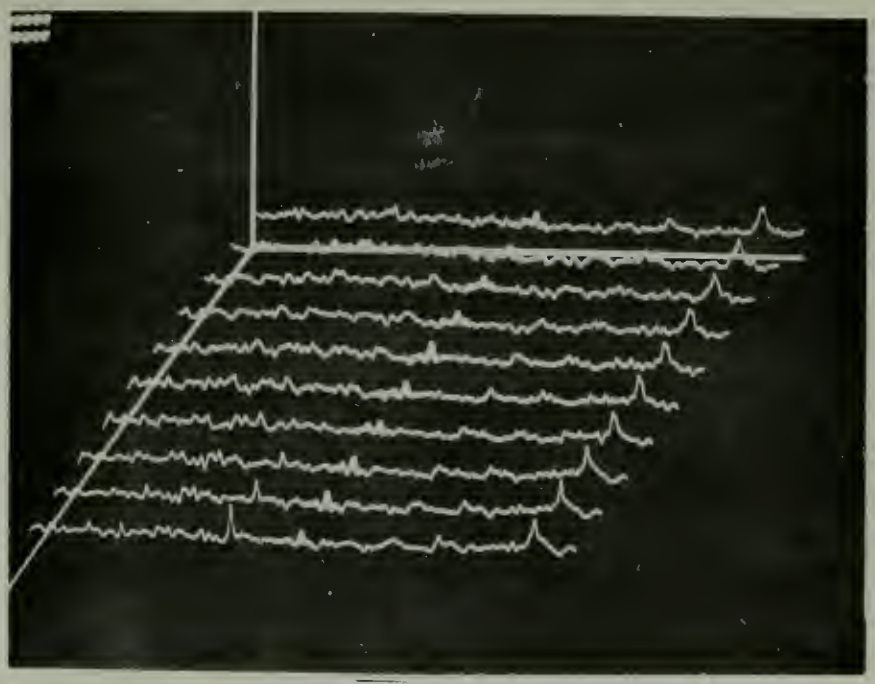
FFT



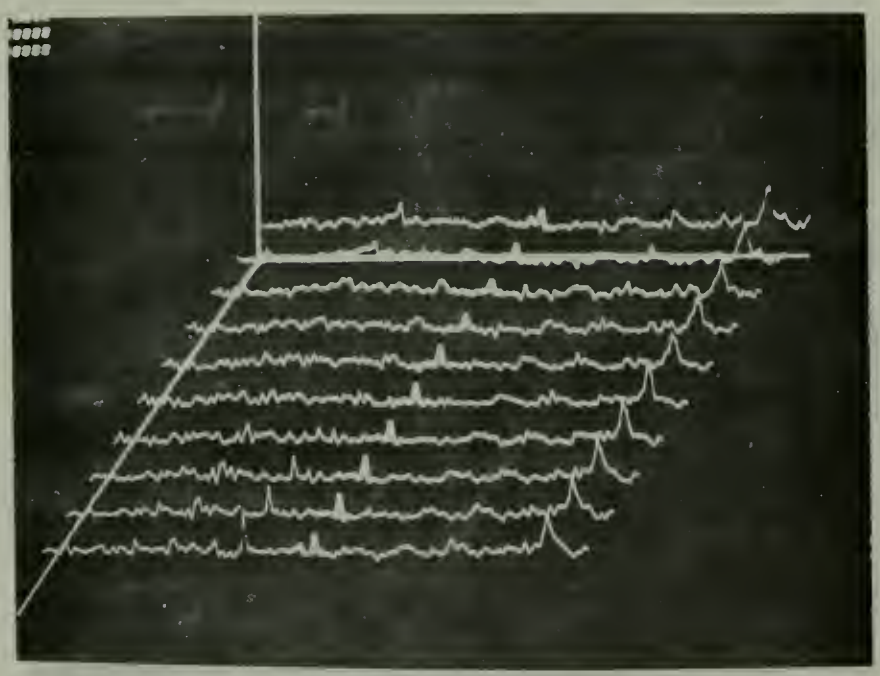
Modified FFT

Underwater Accoustic Data
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 37.5 hz, 0.25 hz resolution

Figure 2-33



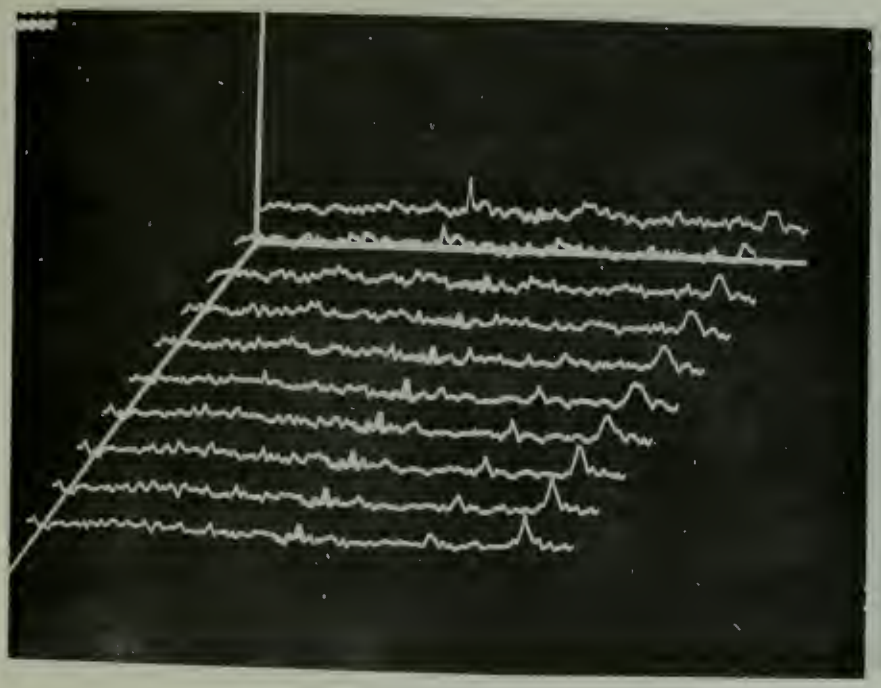
FFT



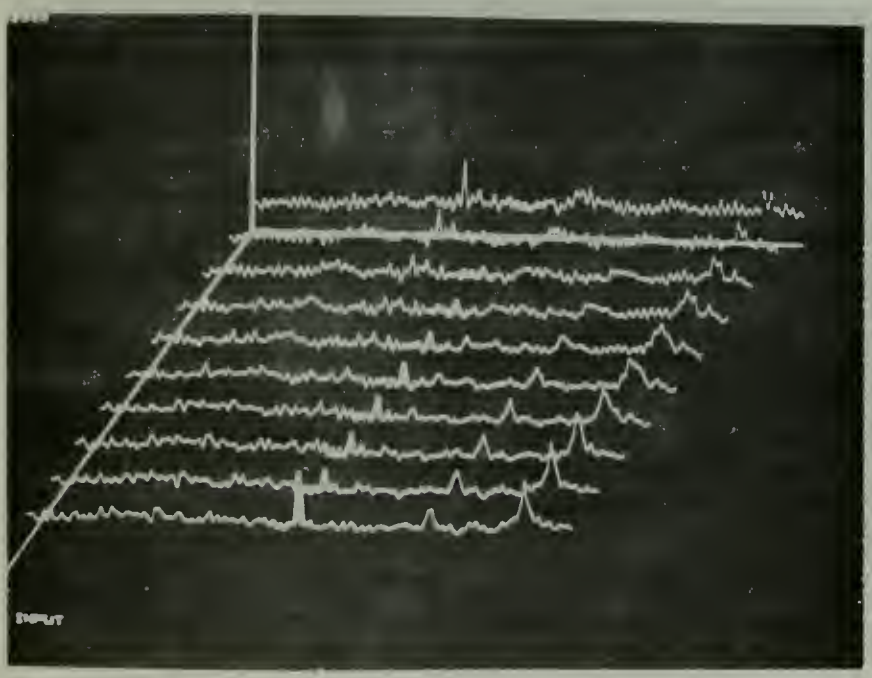
Modified FFT

Underwater Accoustic Data
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 37.5 hz, 0.25 hz resolution

Figure 2-34



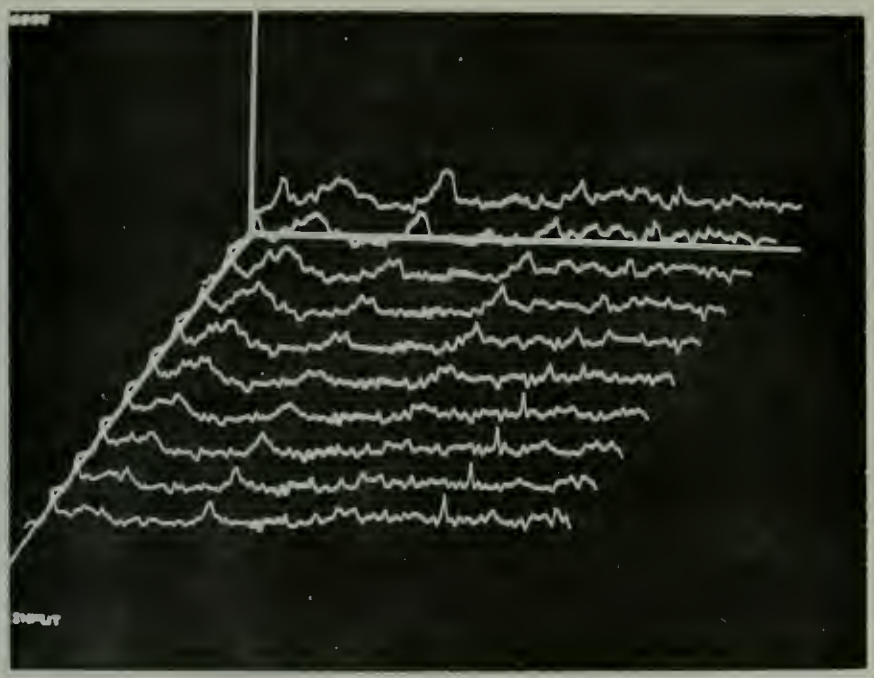
FFT



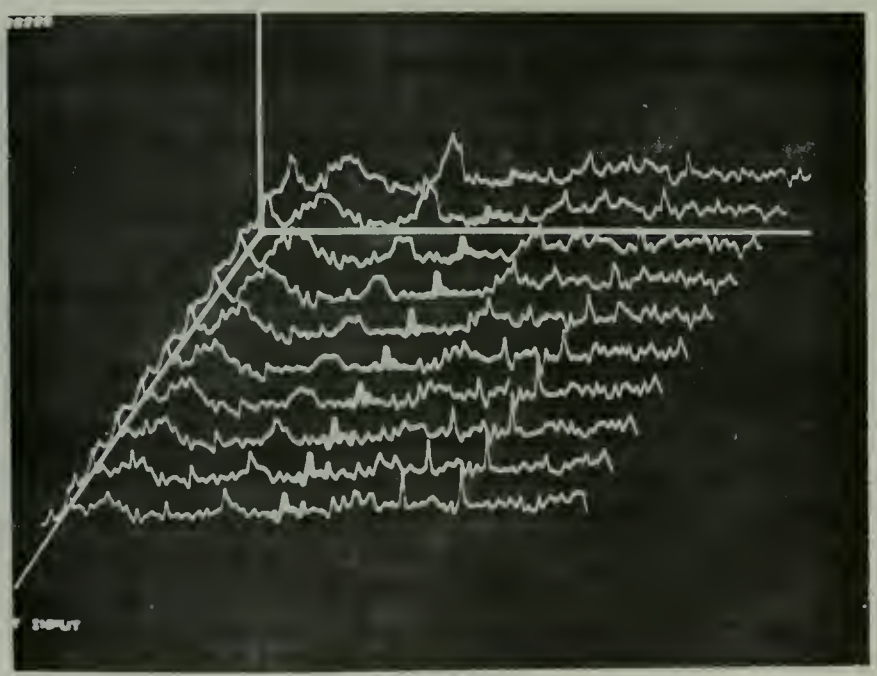
Modified FFT

Underwater Accoustic Data
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 37.5 hz, 0.25 hz resolution

Figure 2-35



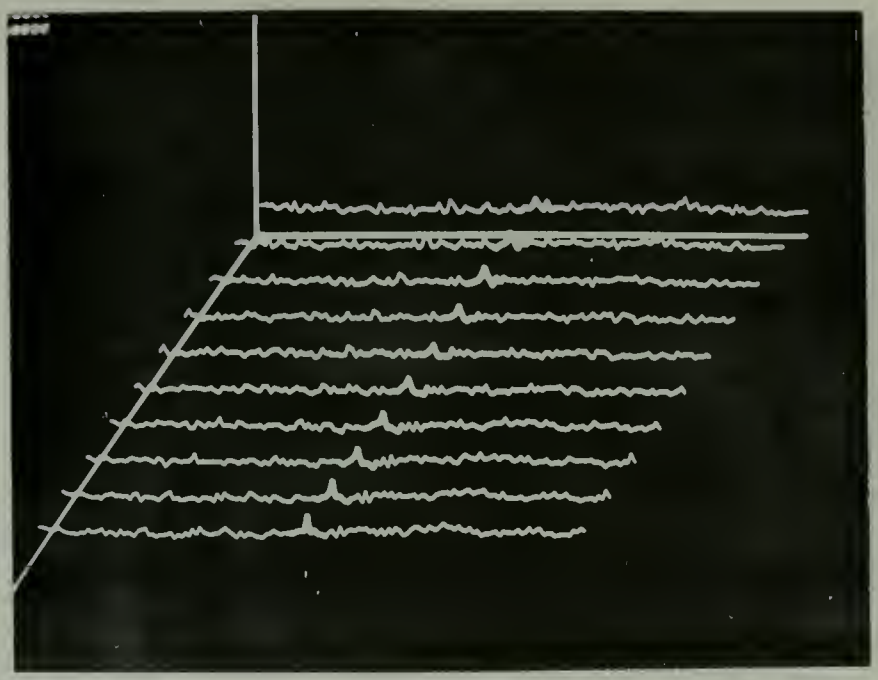
FFT



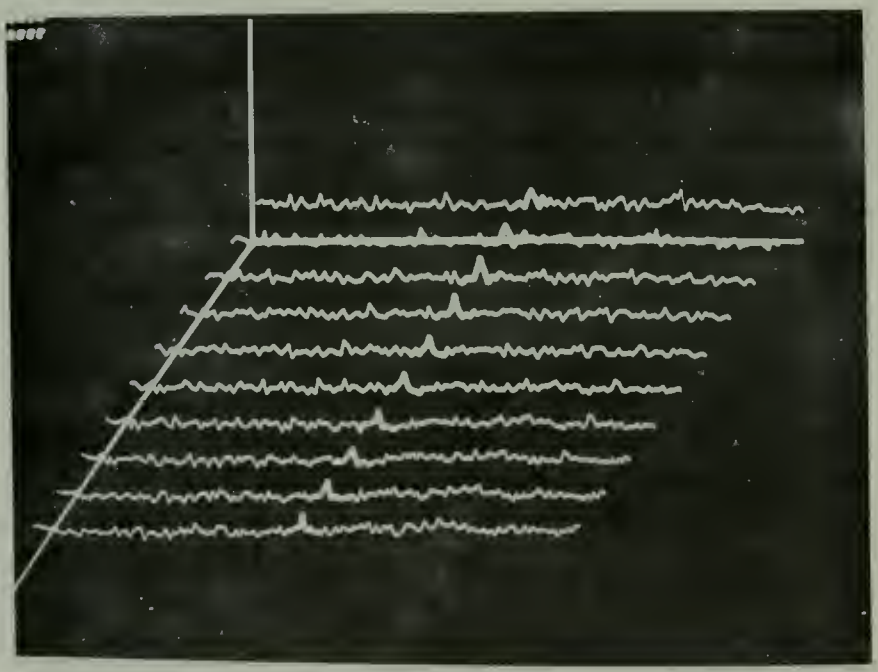
Modified FFT

Underwater Accoustic Data
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 37.5 hz, 0.25 hz resolution

Figure 2-36



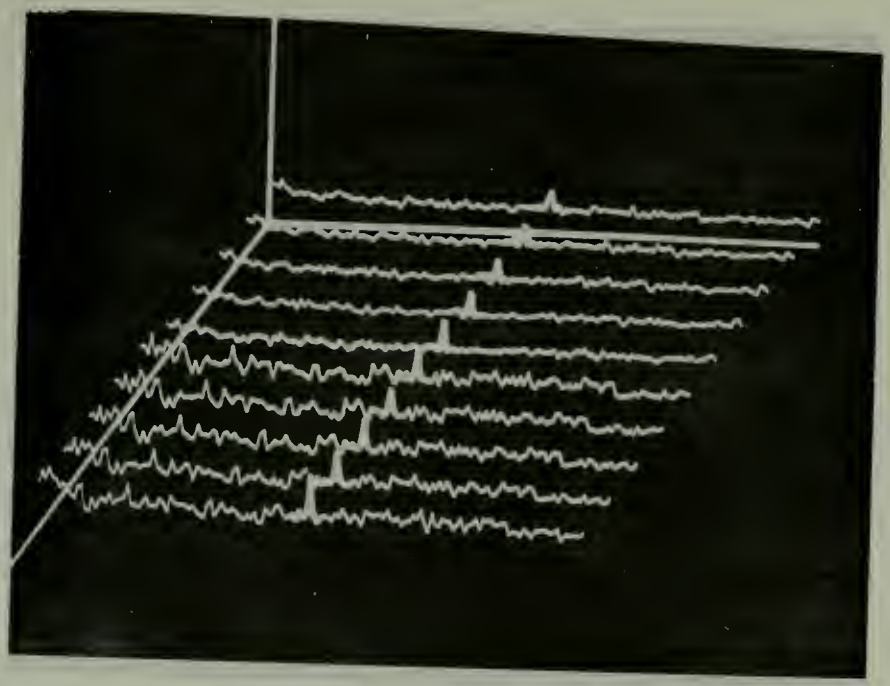
FFT



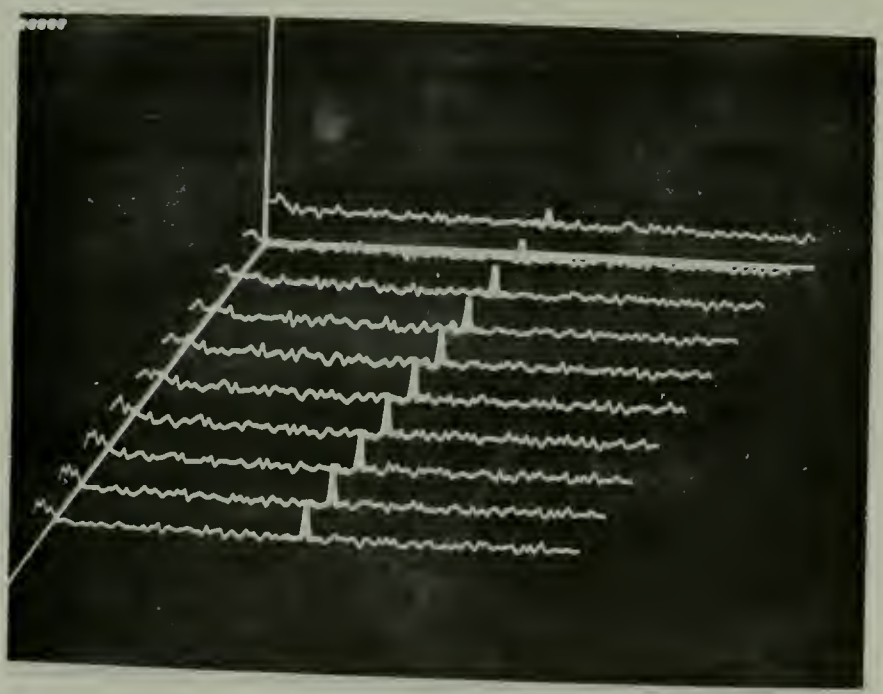
Modified FFT

Underwater Accoustic Data
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 37.5 hz, 0.25 hz resolution

Figure 2-37



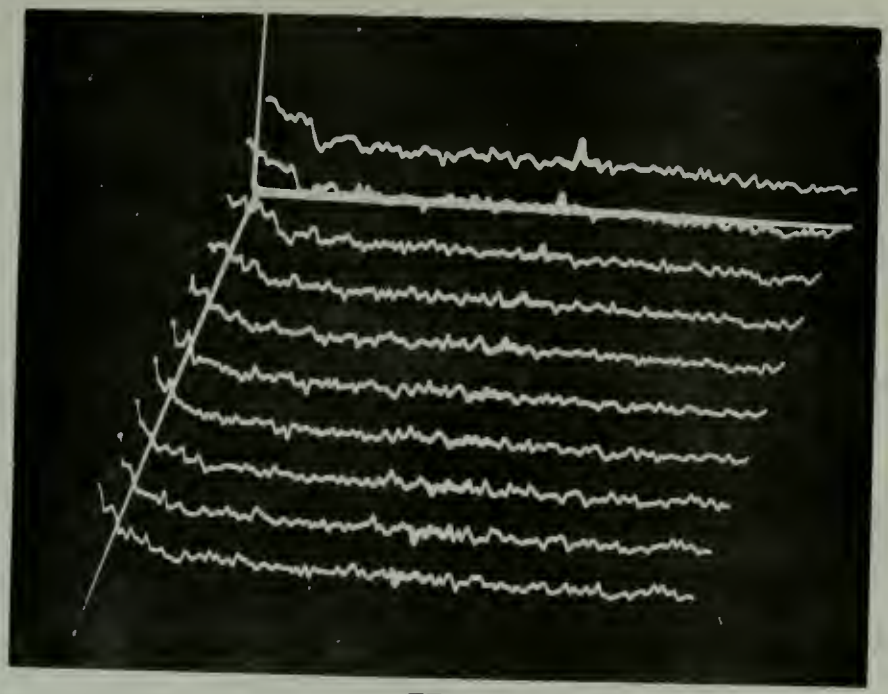
FFT



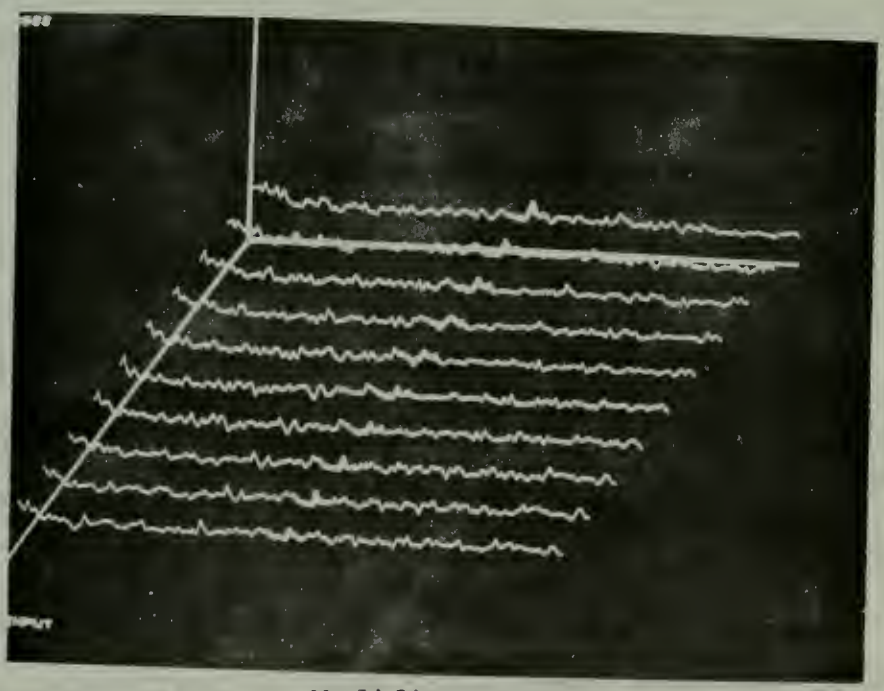
Modified FFT

Underwater Accoustic Data
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 37.5 hz, 0.25 hz resolution

Figure 2-38



FFT



Modified FFT

Underwater Accoustic Data
40 transforms averaged per trace, transform length 2048
Displayed bandwidth 37.5 hz, 0.25 hz resolution

Figure 2-39

IV. PART 3 SUMMARY OF EXPERIMENTAL RESULTS

The data analysed in this paper were basically of three types. A signal or signals with additive Gaussian noise, a signal with additive Gaussian noise plus bursts of Gaussian noise spaced at random intervals with random duration, and finally some underwater acoustic data.

In Section III-B of this investigation, a single sinusoid with additive Gaussian noise was processed with the FFT and the modified FFT. Figures 2-1 through 2-9 indicated that the resulting spectrums from the two transformations are quite similar. The FFT seems to have smoothed the noise slightly better than the modified FFT. In general the FFT seems to provide the better processing gain, but upon close scrutiny of the spectra it is possible to find some traces in which the modified FFT seems to produce a greater signal to noise ratio. The computed S/N output ratios in Table I seem to agree quite well with what is observed in the photographs. The modified FFT seems to outperform the FFT at the large S/N ratios and the FFT dominates the modified FFT at the small signal to noise ratios.

Section III-C of this paper dealt with multiple signal data with additive Gaussian noise. Figures 2-10 through 2-14 depict the output spectra and again it is noted that the spectra are quite similar in overall appearance, with neither transformation appearing to have a distinct advantage over the other. The most interesting results are in

Figures 2-13, where in the FFT spectra the -16 db signal is consistently smaller than the -14 db signal. In the modified FFT spectra the -16 db signal is nearly as large as the -14 db signal. From the -12 db signal on down it is doubtful that one could recognize the signal presented in the figures. Longer transforms or the averaging of more transforms would be required to enhance the signal to noise ratios. Averaging twice as many transforms theoretically increases the S/N output by 3 db and this is verified in Tables II and III.

These tables tend to indicate that the FFT is the better transformation at high S/N ratios and the modified FFT is better at low S/N ratios. For reasons pointed out in Section III-C the -18 db signal should not be used in the analysis.

Some of the apparent inconsistencies in these tables are explained by the fact that different noise coefficients were used in computing the noise statistics for each signal as mentioned in Section III-C. There is no known explanation for the inconsistency of the process gain for the -6 db signal. A search through the adjacent coefficients showed that this coefficient provided the best signal to noise output in the vicinity of the signal for the FFT.

Another interesting aspect of Tables II and III is that in going from a 40 transform average to an 80 transform average the modified FFT S/N ratios are improved more than the FFT S/N ratios.

Section III-D presents data in which bursts of noise were added to the signal in addition to continuous Gaussian noise. In this case the modified FFT consistently outperformed the FFT. The one exception was in the case of the -24 db signal where 40 transforms were averaged. When 80 transforms were averaged the modified FFT outperformed the FFT in every case. Figures 2-25 through 2-28 show the results of the 80 transform average in which it is much easier to believe in the existence of a signal in the case of the modified FFT than in the FFT.

Tables IV and V present the S/N ratios for the spectra. Little needs to be said in this case since the results seem to be consistent with the observations made in Figures 2-18 through 2-28.

Section III-E of this paper presents several signals encountered in the transformation of some actual underwater acoustic data, an area in which signal processing finds a very practical use. The manner of interpretation of the data is very subjective, as is the manner in which it is normally analyzed. In the signals that were presented the modified FFT usually did as well or better than the FFT in making signal presence easily detectable. During all of the research that was done it was found that both transformations generally provided similar results for this type of data. Extremely strong signals were generally enhanced more by the FFT. Medium to weak signals appeared to be about the same in overall detectability for both transformations. Looking

specifically from trace to trace differences were definitely noted and no one particular transformation proved to be consistently better. On some occasions the modified FFT produced signals that the FFT could not, even though later on the signal was produced by the FFT. In the case of Figure 2-29 an analyst confirmed the high probability of a signal being present at the spotlighted frequency. In Figure 2-39 the ability of the modified FFT to retain the signal longer than the FFT is another strong point in favor of the modified FFT but an unusual occurrence. The nonlinearity of the modified FFT with the transformation being a function of the input data itself has made it difficult to draw generalities about its performance except possibly in the case of data similar to that presented in Section III-D of this paper.

V. PART 4 CONCLUSIONS

The data analyzed in this paper is far from conclusive or absolute in terms of allowing any hard and fast conclusions to be drawn. The process does seem to have some merits, especially in the case of the signal plus noise plus burst noise. Whereas this may not appear to be a very practical type of data, it is felt that somewhere between this case and the case of stationary Gaussian noise a practical situation exists in which this transformation may be very useful. Somewhere where strong transients exist, whether they are switching noises, or the explosion of underwater sonic sources, this transformation may be useful if an underlying continuous signal is of interest.

Unexplained are the instances in Section III-E of this paper where the modified FFT was able to bring out signals significantly better than the FFT, or in one case to retain the signal for a longer period of time than the FFT.

On the other hand, the FFT seems better able to distinguish between signal and noise when the noise is stationary Gaussian.

The good characteristics of the transformation do not seem to be dependent on the signal frequency. In Section III-E the signals that were much more evident in the modified FFT than in the FFT, appeared at both ends of the spectrum as well as in the middle of the spectrum. This effect was not limited to one data tape but appeared on two different tapes from different sources.

The main drawback to implementing the modified FFT as it is presently being utilized is the length of time required to perform the transformations, approximately three to four times as long as a similar FFT transformation for 1024 points. However if parallel processing were available the modified FFT could be performed twice as fast as the FFT since each pass only operates on half of the coefficients.

In general there does not seem to be enough difference between the FFT and modified FFT spectra to warrant its use in preference to the FFT. If it is to be used it would seem appropriate to use it when a signal is suspected and the FFT does not reveal it. Also, if the data were known to have characteristics similar to that of the signal plus noise plus burst noise it would warrant strong consideration. One other asset of the transformation that may prove important is the inability of any given output coefficient to exceed some maximum value, determined by the transform length.

With so many variables, such as transform length, frequency resolution, windowing, signal stability, and differing types of background noise and signals, much more investigation will be required to determine the optimum utilization of this transformation. A mathematical and statistical analysis of the modified FFT may give further insight into its best application.

APPENDIX A

FAST FOURIER TRANSFORM

The Fast Fourier Transform is an algorithm for computing the "Discrete Fourier Transform" which evaluates the Fourier Coefficients of the Fourier Series for a discretely sampled waveform.

The term Fast Fourier Transform is somewhat misleading because the FFT is applied to the transformation of periodic functions which do not necessarily satisfy the condition:

$$\int_{-\infty}^{\infty} |f(t)|\, dt < \infty$$

which is required for f(t) (in general) if it is to have a Fourier Transform.

Any discrete analysis of a function f(t) must of necessity be finite in length, say from t=0 to t=T. What happens before t=0 and after t=T is in general not known, or at least the time record in question gives no indication of what was occurring prior to t=0 or what is going to occur after t=T, unless the function is periodic and the time record contains at least one whole period of the waveform. When dealing with the FFT the implied assumption is that the record to be analyzed is one period of a periodic function and that the information contained in this period completely describes the function from $t=-\infty$ to $t=+\infty$.

Periodic functions are more appropriately described by their Fourier Series expansion, rather than by the Fourier

Transform. The term Fast Fourier Transform may be more descriptive if it were renamed the Fast Fourier Series.

A waveform f(t), periodic in t with period T satisfying f(t)=f(t+T) for all t with a finite number of discontinuities on the interval ($0 \leq t \leq T$) can be expanded in a Fourier Series as:

$$f(t) = \sum_{n=-\infty}^{\infty} F(n)\exp[(j2\pi nt)/T]$$

where F(n), the Fourier Coefficients are defined as:

$$F(n) = \frac{1}{T}\int_{0}^{T} f(t)\exp[(-j2\pi nt)/T]dt$$

The above representation of the Fourier Coefficients is for the continuous case. In order to determine the coefficients for the discrete case the integral must be approximated by a summation. Assuming the waveform has been sampled N times in T seconds and that the samples are spaced ΔT seconds apart, then,

$$T=N\Delta T$$

and

$$t=k\Delta T \quad (k=0,1,2,\ldots,N-1)$$

then the sampled version of the Fourier Coefficients can be written:

$$F(n) = \frac{1}{N\Delta T}\sum_{k=0}^{N-1} f(k\Delta T)\exp[(-j2\pi k\Delta T)/T]\Delta T$$

$$=\frac{1}{N}\sum_{k=0}^{N-1} f(k\Delta T)\exp[(-j2\pi nkT)/(NT)]$$

$$=\frac{1}{N}\sum_{k=0}^{N-1} f(k\Delta T)\exp[(-j2\pi nk)/N] \quad (n=0,1,2,\ldots,N-1)$$

The factor 1/N is a scaling factor, useful for obtaining the original amplitude of the input waveform when f(t) is expressed in its Fourier Series. It has no effect on the relative amplitudes of the Fourier Coefficients.

F(n) is the amount of frequency n/T contained in f(t). Abbreviating f(kΔT) as f(k), and omitting the scaling factor 1/N for convenience, then,

$$F(n) = \sum_{k=0}^{N-1} f(k)\exp[(-j2\pi nk)/N]$$

$$= \sum_{k=0}^{N-1} f(k)\{\cos[(2\pi nk)/N]-j\sin[(2\pi nk)/N]\}$$

$$= \sum_{k=0}^{N-1} f(k)\cos[(2\pi nk)/N]-j\sum_{k=0}^{N-1} f(k)\sin[(2\pi nk)/N]$$

which is simply a discrete correlation of f(t) with N sinusoids at the frequencies n/T since,

$$\frac{2\pi nk}{N} = \frac{2\pi tn}{N\Delta T} = \frac{2\pi nt\Delta T}{T\Delta T} = \frac{2\pi nt}{T} \quad \text{where } \frac{n}{T} = n\Delta f$$

F(0) is then the amount of D. C. component contained in f(t) F(1), the amount of frequency 1/T = Δf contained in f(t), F(2), the amount of frequency 2/T = 2Δf contained in f(t), etc., up to n=N-1 for a complex waveform f(t).

If f(t) is real valued the Fourier Coefficient F(N/2-m) = F^*(N/2+m) (where * denotes the complex conjugate).

Proof:

$$F(n) = \frac{1}{N}\sum_{k=0}^{N-1} f(k)\{\cos[(2\pi nk)/N]-j\sin[(2\pi nk)/N]\}$$

let F(n) = R(n)-jI(n), then,

$$R(n) = Re[F(n)] = \frac{1}{N}\sum_{k=0}^{N-1} f(k)\cos[(2\pi nk)/N]$$

for $n=N/2$, $R(N/2)= \frac{1}{N}\sum_{k=0}^{N-1} f(k)\cos(\pi k)= \frac{1}{N}\sum_{k=0}^{N-1}(-1)^k f(k)$

now: $\cos[2\pi k(N/2-m)/N] = \cos[(N\pi k-2\pi km)/N]$

$= \cos[(N\pi k)/N]\cos[(2\pi km)/N] + \sin[(N\pi k)/N]\sin[(2\pi km)/N]$

where the second term is zero for integer k.

$= \cos[(Nk\pi + 2\pi km)/N] = \cos[2\pi k(N/2 + m)/N]$

therefore,

$R(N/2-m) = R(n/2+m)$ $(m=0,1,2,\ldots,N/2-1)$

or the real part of F(n) is symmetric about n=N/2.

A similar proof shows that the imaginary part of F(n) is antisymmetric about n=N/2, i.e.,

$I(N/2-m) = -I(N/2+m)$ $(m=0,1,2,\ldots,N/2-1)$

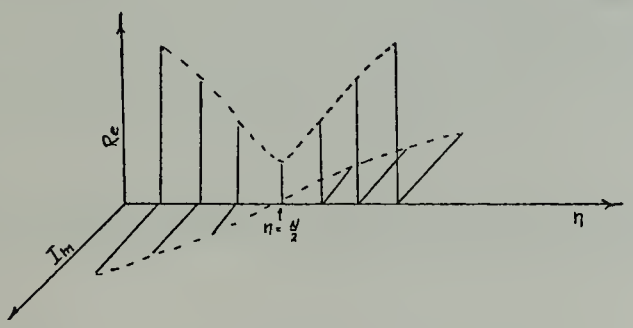


Figure A1-1

That is $F(N/2-m) = F^*(N/2+m)$, implying that all of the spectral information is contained in the first N/2 coefficients, the second half of the spectrum containing redundant information. This is intuitively obvious if one considers that the cosine is an even function and the sine is an odd function.

To more readily visualize the correspondence between the Fourier Transform of a continuous and a sampled waveform, consider a continuous waveform f(t), and its samples counterpart $f_k(t)$ (where the k denotes the sampled version of f(t)). Then $f_k(t)$ can be represented as:

$$f_k(t) = f(k) = \sum_{k=-\infty}^{\infty} f(t)\delta(t-k\Delta T)$$

$$\text{let } d(t) = \sum_{k=-\infty}^{\infty} \delta(t-k\Delta T)$$

$$D(w) \Leftrightarrow d(t)$$

$$F(w) \Leftrightarrow f(t)$$

$$F_k(w) \Leftrightarrow f_k(t)$$

where $\Leftrightarrow$ implies Fourier Transform pairs.

Then $F_k(w) = D(w)*F(w)$

Since $\{\delta(t-k\Delta T)\}$ is a train of impulse functions periodic in t with period ΔT, the Fourier Series representation is used to describe this function, that is:

$$d(t) = \sum_{n=-\infty}^{\infty} D(n)\exp(jwnt) \quad \text{where } w=2\pi f = 2\pi/\Delta T$$

and

$$D(n) = \frac{1}{\Delta T} \int_0^{\Delta T} d(t)\exp(-jwnt)dt$$

then,

$$D(n) = \frac{1}{\Delta T} \int_0^{\Delta T} \sum_{k=-\infty}^{\infty} \delta(t-k\Delta T)\exp(-jwnt)dt$$

$$= \frac{1}{\Delta T} \sum_{k=-\infty}^{\infty} \int_0^{\Delta T} \delta(t-k\Delta T)\exp(-jwnt)dt$$

$$D(n) = \frac{1}{\Delta T}\sum_{k=-\infty}^{\infty} \exp(-jwntk\Delta T) = \frac{1}{\Delta T}\sum_{k=-\infty}^{\infty} \exp[(-j2\pi nk\Delta T)/\Delta T]$$

$$= \frac{1}{\Delta T}\sum_{k=-\infty}^{\infty} \exp(-j2\pi nk) = \frac{1}{\Delta T} \quad \text{for all integer } n,\ k$$

Now,

$$d(t) = \delta(t-k\Delta T) = \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} \exp(jwnt)$$

then:

$$F_k(w) = \int_{-\infty}^{\infty} [f(t)\frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} \exp(jwnt)]\exp(-jwt)dt$$

$$= \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} [f(t)\exp(jwnt)]\exp(-jwt)dt$$

$$= \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} f(t)\exp[j(wn-w)t]dt$$

$$= \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} f(t)\exp[-j(wn-w)t]dt$$

$$= \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty} F(w-wn)$$

The discrete Fourier Transform:

$$F(n) = \frac{1}{N}\sum f(k\Delta T)\exp(-j2\pi nk/N) \quad (n=0,1,2,\ldots,N-1)$$

can be visualized in various ways. As previously mentioned, it can be considered a discrete correlation of the sampled waveform $f(k\Delta T)$ with a series of sinusoids at the frequencies n/T $(n=0,1,2,\ldots,N-1)$ where $T=N\Delta T$ is the length of the input waveform to be transformed, N is the total number of samples, and ΔT is the time between samples.

The time T then determines the frequency resolution of the transformation, that is, $1/T=\Delta f$ or the difference in frequency from one correlating sinusoid to the next. N and T determine the highest frequency of the correlating sinusoids. The ratio of N/T is the sampling rate, i.e., N samples in T seconds. For real valued input functions, the highest correlating frequency is then (N/2-1)/T or half of the sampling rate. Figure A1-2 shows that the waveform must be sampled at a rate at least twice as high as the highest frequency present in f(t) or the spectra of F(w) will overlap. This is called aliasing.

A second visualization is simply that of a matrix operation. Expressing the discrete Fourier Transform in matrix form, omitting the scaling term 1/N, and defining W as exp $(-j2\pi/N)$ then:

$$[F(n)] = [W^{nk}][f(k)]$$

An expansion of these matrices for eight points is given in Figure A1-3a.

Figure A1-3b shows the matrix rewritten with the following substitutions:

$W^{nk} = W^{nkModN}$ since W is periodic with period N.

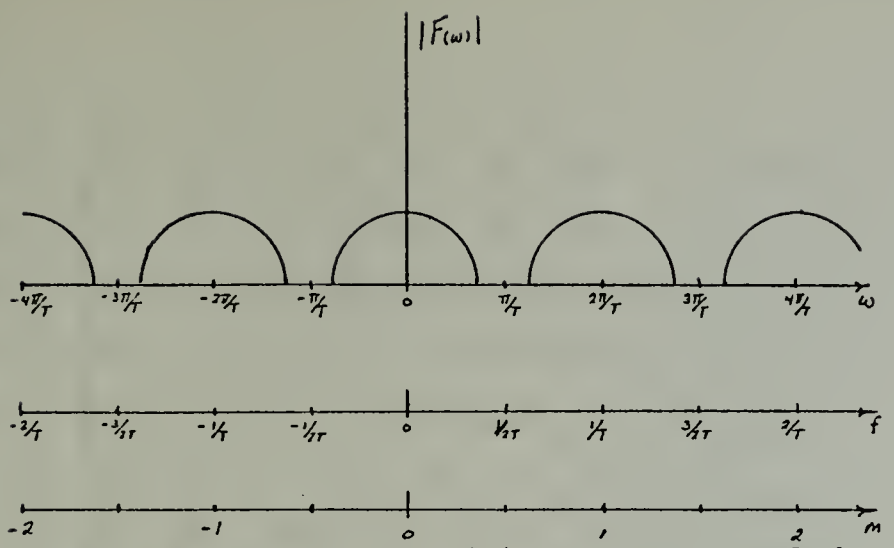
Also, $W^0 = \exp(0) = 1 = -W^4$

$W^1 = \exp(-j\pi/4) = -W^5$

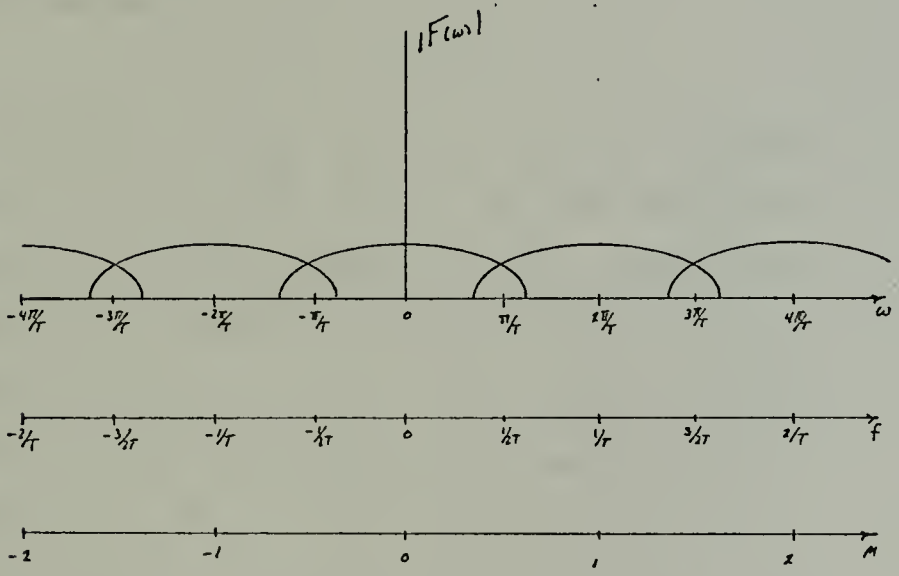
$W^2 = \exp(-j\pi/2) = -W^6$

$W^3 = \exp(-j3\pi/4) = -W^7$

The matrix in Figure A1-3b can be expanded into three matrices as shown in Figure A1-4.



Resulting spectrum if f(t) has been sampled at a rate at least twice as fast as the highest frequency present in f(t).



Resulting spectrum if f(t) has been sampled at a rate slower than twice the highest frequency present in f(t)

Figure A1-2

$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ F(4) \\ F(5) \\ F(6) \\ F(7) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 & W^4 & W^5 & W^6 & W^7 \\ W^0 & W^2 & W^4 & W^6 & W^8 & W^{10} & W^{12} & W^{14} \\ W^0 & W^3 & W^6 & W^9 & W^{12} & W^{15} & W^{18} & W^{21} \\ W^0 & W^4 & W^8 & W^{12} & W^{16} & W^{20} & W^{24} & W^{28} \\ W^0 & W^5 & W^{10} & W^{15} & W^{20} & W^{25} & W^{30} & W^{35} \\ W^0 & W^6 & W^{12} & W^{18} & W^{24} & W^{30} & W^{36} & W^{42} \\ W^0 & W^7 & W^{14} & W^{21} & W^{28} & W^{35} & W^{42} & W^{49} \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix}$$

(a)

Discrete Fourier Transform in matrix form for 8 points

$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ F(4) \\ F(5) \\ F(6) \\ F(7) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & W^1 & W^2 & W^3 & -1 & -W^1 & -W^2 & -W^3 \\ 1 & W^2 & -1 & -W^2 & 1 & W^2 & -1 & -W^2 \\ 1 & W^3 & -W^2 & W^1 & -1 & -W^3 & W^2 & -W^1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -W^1 & W^2 & -W^3 & -1 & W^1 & -W^2 & W^3 \\ 1 & -W^2 & -1 & W^2 & 1 & -W^2 & -1 & W^2 \\ 1 & -W^3 & -W^2 & -W^1 & -1 & W^3 & W^2 & W^1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix}$$

(b)

Matrix in (a) rewritten usin the substitution $W^{nk} = W^{nkModN}$

Figure A1-3

$$\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ F(4) \\ F(5) \\ F(6) \\ F(7) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & W^1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & W^2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & W^3 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -W^1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -W^2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -W^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & W^2 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -W^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & W^2 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -W^2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(4) \\ f(2) \\ f(6) \\ f(1) \\ f(5) \\ f(3) \\ f(7) \end{bmatrix}$$

Expansion of the Discrete Fourier Transform matrix of figure A1-3 into 3 matrices resulting in the Fast Fourier Transform

Figure A1-4

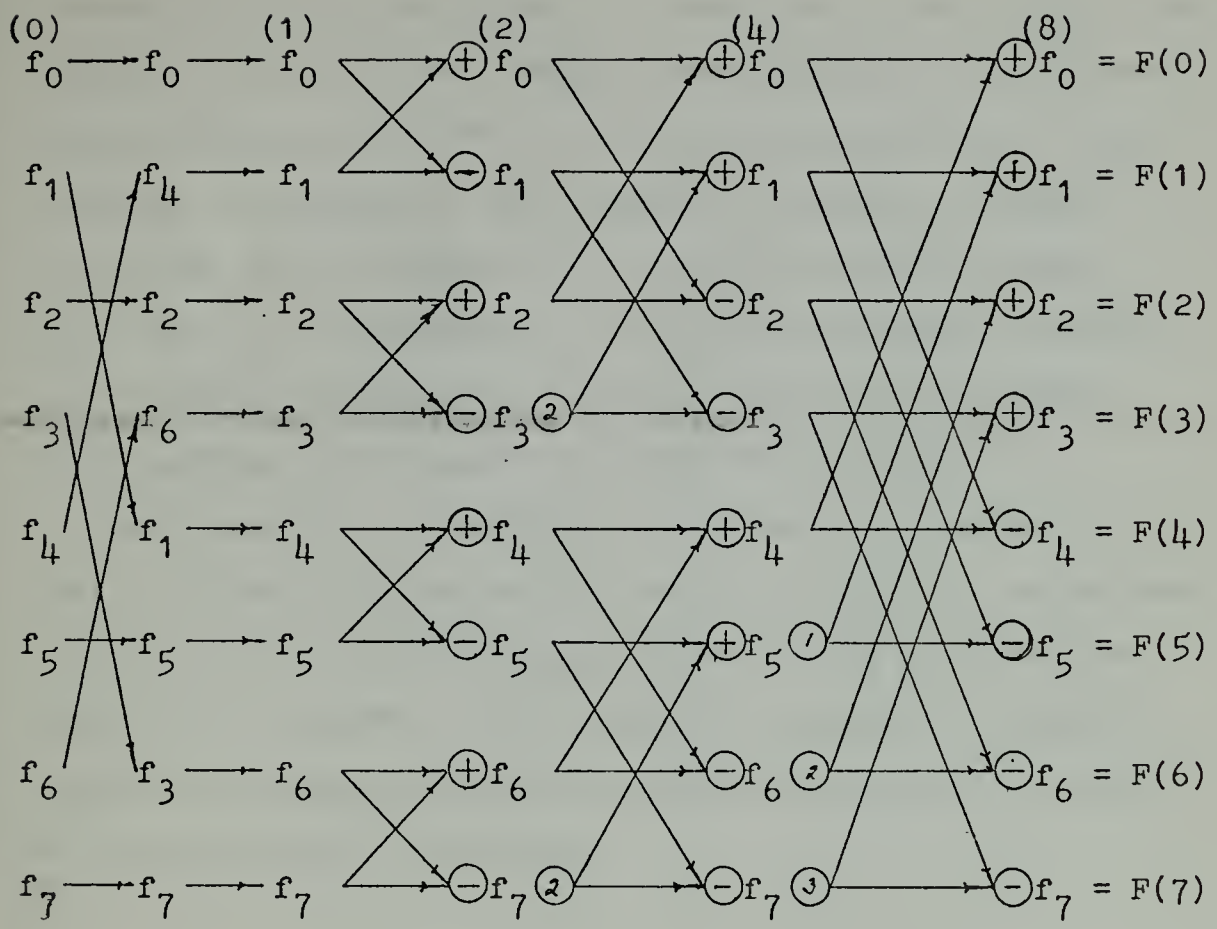
It is this matrix expansion, along with the periodic properties of W^{nk} (the correlating sinusoids), that gives rise to the Fast Fourier Transform. In general, if a sampled input waveform has N samples where $N=2^M$ and M is a positive integer, then the N×N matrix of operators (the matrix of W's) can be expanded into a product of M matrices with the stipulation that the input vector is bit reversed (note the reordering of the input vector). This stipulation is not a requirement but it is the method that is used throughout this study. The process of bit reversal allows the matrix multiplications to be performed one after the other without a requirement to save any intermediate results. The matrix expansion can be performed in other ways such that the input vector is left in its natural order and the output vector is in bit reversed order, or possibly the odd harmonics of Δf are grouped together and the even harmonics are grouped together. It is also possible to perform the operations without bit reversing the input or the output vectors but then a requirement exists for the retention of intermediate results.

Bit reversal is simply the reversing of the order of the digits that make up a number in binary arithmetic. In the above situation N=8, and M=3. For example, 110 (6) becomes 011 (3) when its bits are reversed. The effect of bit reversal is to place each element in the input vector next to an element that is N/2 elements away from it, i.e., N/2=4 in this case so element 1 is placed next to element 5, 2 next to 6, etc.

To go one step further, the matrices of Figure A1-4 can be further reduced by using a tree graph or what is commonly called an "FFT butterfly". The expanded matrices have many zero elements, in fact each matrix contains only 2N elements that are non-zero and a large portion of these are 1's or -1's. The original Discrete Fourier Transform required N^2 complex operations in its execution; the expanded matrices of the Fast Fourier Transform only have $N\log_2 N$ operations and thus the term "Fast".

The matrices have been rewritten in butterfly format in Figure A1-5. The column labeled "0" is the input vector. The column labeled "1" is the bit reversed form of the input vector. The column labeled "2" is the results of the first matrix multiplication, or the results after the first pass through the algorithm. The number 2 is used to describe this column since two of the original elements make up each of the elements at this point in the algorithm. The column labeled "4" is the result of the second matrix multiplication, or the result after the second pass through the algorithm. As above the number 4 is used to describe this column since four of the original elements make up each of the elements at this point in the algorithm. The nomenclature is similar for each of the columns in the butterfly.

When two arrows meet at a circle with a + or - in it, then addition or subtraction is to take place. For example, in the column labeled 2 element f_0 is made up of elements $f_0 + f_1$ from column 1; element f_1 in column 2 is made up of



FFT butterfly or flow diagram for an 8 point transformation

Figure A1-5

elements f_0 - f_1 from column 1. The circles that contain an argument like 1, 2, or 3 imply a multiplication of the preceding element by W^1, W^2, or W^3 before an addition or subtraction is performed. For example, in column 4 element f_0 is made up of elements $f_0 + f_2$ from column 2, element f_1 is made up of elements $f_1 + f_3W^2$ from column 2, element f_2 is made up of elements $f_0 - f_2$ from column 2, element f_3 is made up of elements $f_1 - f_3W^2$ from column 2.

The butterfly is identical to the matrix approach in that each pass through the butterfly is identical to performing one of the matrix multiplications in the matrix representation of the FFT. The butterfly condenses the matrix approach by elimination of the zeros in each matrix leaving only the necessary operations.

```
      SUBROUTINE TXPH(D,E,C,S,IBUF,NT,MPT,ISINE)
      DIMENSION D(1),E(1),C(1),S(1),IBUF(1)
C*****************************************************************
C                                                                 *
C     THIS SUBROUTINE WHEN USED IN CONJUNCTION WITH THE SUBROUTINE*
C     BFPTM WILL PERFORM THE MODIFIED FAST FOURIER TRANSFORM.  IT  *
C     PERFORMS ONE TRANSFORM ON ALL OF THE DATA TO OBTAIN THE     *
C     COEFFICIENTS RELATED TO THE ODD HARMONICS OF THE FUNDAMENTAL*
C     FREQUENCY OF THE TRANSFORM (FO).  IT THEN DIVIDES THE DATA IN*
C     HALF AND PERFORMS TWO TRANSFORMS TO OBTAIN THE COEFFICIENTS *
C     4NFO-2 (N=1,2,3,...).  IT THEN DIVIDES THE DATA INTO QUARTERS*
C     AND PERFORMS FOUR TRANSFORMS TO OBTAIN THE COEFFICIENTS 8NFO-4*
C     (N=1,2,3,...) ETC., UNTIL ALL OF THE COEFFICIENTS HAVE BEEN *
C     OBTAINED .  IT ALSO ORDERS AND ADDS THE OUTPUTS OF THE TRANSFORMS*
C     TO RETURN THE OUTPUT COEFFICIENTS IN THE CORRECT ORDER.  THE*
C     OUTPUT COEFFICIENTS ARE RETURNED IN ARRAYS D AND E.         *
C     THE SUBROUTINE IS SET UP TO RECIEVE THE INPUT DATA IN INTEGER*
C     FORM.  IF REAL DATA IS TRANSFORMED THE ARRAY IBUF SHOULD BE *
C     REPLACED BY A REAL ARRAY.                                   *
C     THE CALLING PARAMETERS ARE:                                 *
C       D - STORAGE ARRAY FOR THE REAL COMPONENTS OF THE COEFFICIENTS*
C       E - THE STORAGE ARRAY FOR THE IMAGINARY COMPONENTS OF THE  *
C           COEFFICIENTS                                          *
C       C - STORAGE FOR THE COSINE TABLE                          *
C       S - STORAGE FOR THE SINE TABLE                            *
C       IBUF - INPUT DATA ARRAY (DATA TO BE TRANSFORMED)          *
C       NT - THE TOTAL NUMBER OF POINTS TO BE TRANSFORMED         *
C       MPT - THE POWER OF 2 SUCH THAT 2**MPT=NT                  *
C       ISINE - IF ISINE=0 THIS SUBROUTINE WILL COMPUTE THE SINE AND*
C               COSINE TABLES.  IF ISINE=1 THE SINE AND COSINE TABLES*
C               SHOULD HAVE BEEN PRECOMPUTED.                     *
C                                                                 *
C*****************************************************************
C                                                                 *
C     INITIALIZE FOR FIRST PASS THROUGH ALGORITHM                 *
```

```
*                                                                      *
      N=NT
      MP=MPT
      NT2=NT/2
      NT4=NT2/2
      N2=N/2
      MPTM1=MPT-1
*                                                                      *
*     COMPUTE SINE/COSINE TABLE                                        *
*                                                                      *
      IF(ISINE.EQ.1) GO TO 25
      PI=3.1415926535
      DO 5 J=1,NT4-1
      S(J)=SIN(PI*J/NT2)
      C(J)=COS(PI*J/NT2)
    5 CONTINUE
*                                                                      *
*     COMMENCE TRANSFORMATION OF DATA                                  *
*     WHEN K=1 THE ODD HARMONICS OF THE FUNDAMENTAL FREQUENCY OF THE   *
*     TRANSFORM (FO) WILL BE COMPUTED, IE, NFO (N=1,3,5,...,((NT/2)-1)).*
*     WHEN K=2 THE COEFFICIENTS 2NFO (N=1,3,5,...,((NT/4)-1)) WILL     *
*     BE COMPUTED.  WHEN K=3 THE COEFFICIENTS 4NFO (N=1,3,5,...,       *
*     ((N/8)-1)) WILL BE COMPUTED, ETC.                                *
*                                                                      *
   25 DO 90 K=1,MPTM1
*                                                                      *
*     NITRAN CONTROLS THE TRANSFORM LENGTH.  NITRAN=1 IMPLIES AN NT    *
*     POINT TRANSFORM WILL BE PERFORMED YIELDING THE COEFFICIENTS      *
*     ASSOCIATED WITH K=1.  NITRAN=2 IMPLIES TWO NT/2 POINT TRANSFORMS *
*     WILL BE PERFORMED YIELDING THE COEFFICIENTS ASSOCIATED WITH K=2.  *
*     NITRAN=4 IMPLIES FOUR NT/4 POINT TRANSFORMS WILL BE PERFORMED    *
*     YIELDING THE COEFFICIENTS ASSOCIATED WITH K=3, ETC.              *
*                                                                      *
      NITRAN=2**(K-1)
*                                                                      *
```

```
*     COMPUTE OUTPUT COEFFICIENTS BY LEVEL.  L CONTROLS THE NUMBER     *
*     OF TRANSFORMS PERFORMED FOR EACH K.  INDX SEPARATES THE INPUT    *
*     DATA ARRAY (IBUF) INTO NITRAN EQUAL SEGMENTS.                    *
*                                                                      *
      DO 80 L=1,NITRAN
      INDX=(L-1)*N
      DO 50 J=1,N
      D(J)=IBUF(INDX+J)/(2.0**23)
      E(J)=0.0
   50 CONTINUE
*                                                                      *
*     TRANSFORM DATA                                                   *
*                                                                      *
      CALL BFPTM(D,E,C,S,NT,N,MP)
*                                                                      *
*     INITIALIZE FOR THE REINDEXING OF THE N POINT TRANSFORM OUTPUT    *
*     COEFICIENTS TO COINCIDE WITH THE NT POINT TRANSFORM.             *
*                                                                      *
      LSTEP2=2**(K-1)
      LONE=LSTEP2+1
*                                                                      *
*     ON FIRST PASS THROUGH THE ALGORITHM ZERO UPPER HALF OF THE D     *
*     AND E ARRAYS FOR TEMPORARY STORAGE OF THE OUTPUT COEFFICIENTS.   *
*     AFTER THE FIRST PASS ALL REMAINING TRANSFORMS WILL BE OF LENGTH  *
*     NT/2 OR LESS AND THE UPPER HALF OF THE D AND E ARRAYS WILL NO    *
*     LONGER BE REQUIRED FOR TRANSFORMATION PURPOSES                   *
*                                                                      *
      IF(K.NE.1) GO TO 70
      DO 65 J=N2+1,N
      D(J)=0.0
      E(J)=0.0
   65 CONTINUE
*                                                                      *
*     PLACE N POINT TRANSFORM OUTPUT COEFFICIENTS INTO THE NT POINT    *
*     TRANSFORM OUTPUT COEFFICIENT ARRAY.                              *
```

```
*                                                                      *
   70 DO 75 J=2,N2,2
      JDEX=LONE+LSTEP2*(J-2)+NT2
      D(JDEX)=D(J)+D(JDEX)
      E(JDEX)=E(J)+E(JDEX)
   75 CONTINUE
   80 CONTINUE
*                                                                      *
*     REDUCE N AND MP TO CORRESPOND WITH THE PROPER TRANSFORM LENGTH   *
*     FOR THE NEXT PASS THROUGH THE ALGORITHM.                         *
*                                                                      *
      N=N/2
      N2=N/2
      MP=MP-1
   90 CONTINUE
*                                                                      *
*     MOVE OUTPUT COEFFICIENTS FROM THE UPPER HALF OF D AND E ARRAYS   *
*     TO LOWER HALF OF D AND E ARRAYS AND RETURN.                      *
*                                                                      *
      DO 100 J=1,NT2
      I=NT2+J
      D(J)=D(I)
      E(J)=E(I)
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BFPTM(D,E,C,S,NT,N,MP)
      DIMENSION D(1),E(1),C(1),S(1)
*****************************************************************
*                                                               *
*     THIS SUBROUTINE BIT REVERSES AND PERFORMS THE MODIFIED VERSION   *
*     OF THE FFT.  IT ONLY COMPUTES THE ODD COEFFICIENTS (THE ODD      *
*     HARMONICS OF THE FUNDAMENTAL FREQUENCY OF THE TRANSFORM).  AFTER *
*     PASS 2 THROUGH THE ALGORITHM EACH ODD COEFFICIENT IS DIVIDED BY  *
*     ITS MAGNITUDE, (THE MODIFICATION).                               *
*     THE CALLING PARAMETERS ARE:                                      *
*       D - THE INPUT ARRAY TO BE TRANSFORMED                          *
*       E - THE STORAGE ARRAY FOR THE IMAGINARY COMPONENTS OF THE      *
*           COEFFICIENTS                                               *
*       (ARRAYS D AND E CONSTITUTE THE REAL AND IMAGINARY PARTS OF THE *
*       COEFFICIENTS.  THE OUTPUT COEFFICIENTS ARE RETURNED IN ARRAYS  *
*       D AND E)                                                       *
*       C - STORAGE FOR THE PRECOMPUTED COSINE TABLE                   *
*       S - STORAGE FOR THE PRECOMPUTED SINE TABLE                     *
*       NT - THE TOTAL NUMBER OF POINTS TO BE TRANSFORMED              *
*       N - THE TRANSFORM LENGTH (IF NT=16, AND N=16, THEN THE ODD     *
*           COEFFICIENTS 1,3,5,7 WILL BE RETURNED IN D AND E.  IF N=8,  *
*           THEN THE ODD COEFFICIENTS OF THE 8 POINT TRANSFORM WILL BE *
*           RETURNED, BUT THESE WILL BE COEFFICIENTS 2 AND 6 OF THE    *
*           ORIGINAL 16 POINT TRANSFORM.  IF N=4, COEFFICIENT NUMBER 4 *
*           OF THE ORIGINAL 16 POINT TRANSFORM WILL BE RETURNED.)      *
*       MP - THE POWER OF 2 SUCH THAT 2**MP=N.                         *
*                                                                      *
*****************************************************************
      N2=N/2
*                                                                      *
*     BIT REVERSAL                                                     *
*                                                                      *
      J=1
      DO 60 I2=1,N
      IF(J-I2)10,30,30
```

```
   10 TEMP=D(I2)
      D(I2)=D(J)
      D(J)=TEMP
   30 M=N2
   40 IF(J-M)60,60,50
   50 J=J-M
      M=M/2
      IF(M-1)60,40,40
   60 J=J+M
*                                                                      *
*     FIRST PASS OF THE ALGORITHM (SUM AND DIFFERENCE)                 *
*                                                                      *
      DO 200 J=1,N,2
      D(J+1)=D(J)-D(J+1)
  200 CONTINUE
*                                                                      *
*     SECOND PASS AND THE FFT MODIFICATION                             *
*                                                                      *
      DO 300 J=2,N,4
      E(J)=D(J+2)
*                                                                      *
*     DIVISION OF THE COEFFICIENTS BY THEIR MAGNITUDE                  *
*                                                                      *
      DIV=SQRT(D(J)*D(J)+E(J)*E(J))
*                                                                      *
*     PREVENT DIVISION BY ZERO                                         *
*                                                                      *
      IF(DIV.LE.0.0001) GO TO 250
      D(J)=D(J)/DIV
      E(J)=E(J)/DIV
  250 D(J+2)=D(J)
      E(J+2)=-E(J)
  300 CONTINUE
*                                                                      *
*     THIRD AND REMAINING PASSES OF THE ALGORITHM                      *
```

```
*                                                                       *
      DO 1000 K=3,MP,1
      INDX4=2**(K-2)
      INDX2=2*INDX4
      INDX=2*INDX2
      DO 750 J=1,N,INDX
      DO 500 I=2,INDX4,2
      K2=J+I-1
      K1=K2+INDX2
      K3=J+INDX2+1-I
      K4=K3+INDX2
*                                                                       *
*     L IS THE INDEX FOR THE PRECOMPUTED SINE AND COSINE TABLE          *
*                                                                       *
      L=NT*(I-1)/INDX
      TEMPR=D(K1)*C(L)-E(K1)*S(L)
      TEMPI=D(K1)*S(L)+E(K1)*C(L)
      D(K1)=D(K2)-TEMPR
      E(K1)=E(K2)-TEMPI
      D(K2)=D(K2)+TEMPR
      E(K2)=E(K2)+TEMPI
      D(K3)=D(K1)
      E(K3)=-E(K1)
      D(K4)=D(K2)
      E(K4)=-E(K2)
  500 CONTINUE
  750 CONTINUE
 1000 CONTINUE
 9000 RETURN
      END
```

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Libaray, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor G. A. Rahe, Code 72Ra Computer Science Group Naval Postgraduate School Monterey, California 93940	4
4.	LT Robert E. Beal, USN Air Anti-Submarine Squadron FORTY ONE Naval Air Station, North Island San Diego, California 92135	1
5.	Professor D. M. Powers, Code 72Pw Computer Science Group Naval Postgraduate School Monterey, California 93940	1
6.	Department Chariman, Code 57 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1

153425

Thesis
B2856 Beal
c.1

An investigation of a method for obtaining a numerical indicator of phase coherence.